DOCUMENT RESUME

ED 150 955 IR 005 492

AUTHOR	Lehnert, Wendy
TITLE	The Process of Question Answering. Research Report No. 88.
SPONS AGENCY	Advanced Research Projects Agency (DOD), Washington, D.C.
PUB DATE	May 77
CONTRACT	N00014-75-C-1111
NOTE	293p.; Ph.D. Dissertation, Yale University
EDRS PRICE	MF-$0.83 HC-$15.39 Plus Postage.
DESCRIPTORS	*Computer Programs; Computers; *Conceptual Schemes; *Information Processing; *Language Classification; *Models; Program Descriptions
IDENTIFIERS	*QUALM; *Question Answering Systems

ABSTRACT

The computational model of question answering proposed by a computer program, QUALM, is a theory of conceptual information processing based on models of human memory organization. It has been developed from the perspective of natural language processing in conjunction with story understanding systems. The processes in QUALM are divided into four phases: (1) conceptual categorization; (2) inferential analysis; (3) content specification; and (4) retrieval heuristics. QUALM provides a concrete criterion for judging the strengths and weaknesses of story representations. As a theoretical model, QUALM is intended to describe general question answering, where question answering is viewed as a verbal communication device between people. (Author/KP)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED150955

This work was presented to the Graduate School of Yale University in candidacy for the degree of Doctor of Philosophy.

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

THE PROCESS OF QUESTION ANSWERING

May 1977

Research Report #88

Wendy Lehnert

This work was supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored under the Office of Naval Research under contract N00014-75-C-1111.

IR005492

ABSTRACT

Problems in computational question answering assume a new perspective when question answering is viewed as a problem in natural language processing. A theory of question answering has been proposed from this viewpoint which relies on ideas in conceptual information processing and theories of human memory organization. This theory of question answering has been implemented in a computer program, QUALM. QUALM is currently used by two story understanding systems (SAM and PAM) to complete a natural language processing system which reads stories and answers questions about what was read.

The processes in QUALM are divided into four phases: (1) Conceptual Categorization, (2) Inferential Analysis, (3) Content Specification, and (4) Retrieval Heuristics. Conceptual Categorization guides subsequent processing by dictating which specific inference mechanisms, and memory retrieval strategies should be invoked in the course of answering a question. Inferential Analysis is responsible for understanding what the questioner really meant when a question should not be taken literally. Content Specification determines how much of an answer should be returned in terms of detail and elaborations. Retrieval Heuristics do the actual digging in order to extract an answer from memory. All of the inference processes within these four phases are independent of language, operating within conceptual representations.

QUALM represents a theory of question answering which is motivated by theories of natural language processing. Within the context of story understanding, QUALM has provided a concrete criterion for judging the strengths and weaknesses of story representations generated by SAM and PAM. If a system understands a story, it should be able to answer questions about that story in the same way that people do. Although the computer implementation of QUALM is currently limited to the application of answering questions about stories, the theoretical model goes beyond this particular context. As a theoretical model QUALM is intended to describe general question answering, where question answering in its most general form is viewed as a verbal communication device between people.

PREFACE

When a person understands a story, he can demonstrate his understanding by answering questions about the story. Since questions can be devised to query any aspect of text comprehension, the ability to answer questions is the strongest possible demonstration of understanding. Question answering is therefore a task criterion for evaluating reading skills.

If a computer is said to understand a story, we must demand of the computer the same demonstrations of understanding that we require of people. Until such demands are met, we have no way of evaluating text understanding programs. Any computer programmer can write a program which inputs text. If the programmer assures us that his program 'understands' text, it is a bit like being reassured by a used car salesman about a suspiciously low speedometer reading. Only when we can ask a program to answer questions about what it reads will we be able to begin to assess that program's comprehension.

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor, Professor Roger Schank, whose creativity and teaching style have transformed the problems of natural language processing into intensely challenging and gratifying research efforts. He taught me to conduct research with a delicate balance of disciplined restraint and inspired abandon. But lest we get too carried away, it also helps to be slightly manic and to have a big computer.

Special thanks go to Professor Robert Abelson, social psychologist, member of the Yale A. I. Project, and an endless source of ideas that tend to be about ten years ahead of their time.

Dr. Christopher Riesbeck's ideas and intuitions have been another major and valuable influence. His remarkable sense of humor counterbalances the pressures of impossible deadlines and other bothersome distractions for everyone who has the pleasure of working with him.

The preparation of this manuscript was greatly aided by Dr. Drew McDermott who read draft version 2.5 of his own free will. I am grateful for his many extensive and thoughtful comments which facilitated the final rewriting (polishing is not quite the word) of this thesis.

I want to thank Professor Alan Perlis for being on my reading commitee along with Professors Robert Abelson and Roger Schank.

I would also like to acknowledge the cordial efforts of Dr. Daniel Bobrow who invited me to spend a summer at Xerox PARC while he and Terry Winograd collaborated on KRL. While we did not always agree on issues in natural language processing, our arguments were often enlightening, and the opportunity to experience KRL first hand was both stimulating and instructive. One of the computer programs (COIL) described in this thesis was written in an experimental implementation of KRL during my stay at Xerox.

The other members of the Understander Group at Xerox PARC all contributed to making my stay at Xerox a pleasant and productive experience: Ronald Kaplan, Martin Kay, David Levy, Paul Martin, and Henry Thompson. Special thanks go to David Levy, Paul Martin, and Henry Thompson for patiently helping me unravel the mysteries of INTERLISP and KRL.

I am especially grateful to all the past and present student members of the Yale A.I. Project who have made being a graduate student much more fun than it's supposed to be: Jaime Carbonell, Richard Cullingford, Gerald DeJong, Anatole Gershman, Richard Granger, Leila Habib, James Meehan, Richard Proudfoot, Mallory Selfridge, Walter Stutzman, and Robert Wilensky.

Finally, I must thank my parents who have held up rather admirably throughout all my successive childhoods.

And a very special word of appreciation goes to my husband, Richard Goldstein, for convincing me that two free spirits are better than one.

This work was supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored under the Office of Naval Research under contract N00014-75-C-1111.

TABLE OF CONTENTS

ABSTRACT
PREFACE ii
ACKNOWLEDGEMENTS iii
TABLE OF CONTENTS v

CHAPTER 1: PROBLEMS, PREVIEWS AND PROGRAMS

1.0 Introduction 1
1.1 What's Hard About Answering Questions 1
    1.1.1 Conceptual Analysis 1
    1.1.2 Reference Recognition 2
    1.1.3 Conceptual Categorization 3
    1.1.4 Understanding in Context 3
    1.1.5 Conversational Interpretation 4
    1.1.6 Inferences 5
    1.1.7 Knowledge-Based Interpretation 6
    1.1.8 Memory-Based Interpretation 7
    1.1.9 Selecting the Best Answer 8
    1.1.10 State Descriptions 9
    1.1.11 Content Specification 11
    1.1.12 Accessing Memory 12
    1.1.13 Constructing Information 13
    1.1.14 Finding Only What's Needed 14
1.2 Where We're Going 15
    1.2.1 How to Survive the Trip 17
1.3 QUALM 17
1.4 Computer Programs Using QUALM 18
    1.4.1 SAM 19
    1.4.2 PAM 25
    1.4.3 ASP 26
    1.4.4 COIL 28

PREFACE TO INTERPRETATION: UNDERSTANDING QUESTIONS

0. Introduction 30
1. Four Levels of Understanding 30
2. The Conceptual Parse 31
3. Memory Internalization 32
4. Conceptual Categorization 34
5. Inferential Analysis 34

CHAPTER 2: CONCEPTUAL CATEGORIES FOR QUESTIONS

2.0 Introduction 37
2.1 Causal Antecedent 42
2.2 Goal Orientation 44
2.3 Enablement 46
2.4 Causal Consequent 49
2.5 Verification 52
2.6 Disjunctive 54
2.7 Instrumental/Procedural 56
2.8 Concept Completion 59
2.9 Expectational 61
2.10 Judgemental 62
2.11 Quantification 64
2.12 Feature Specification 66
2.13 Request 67
2.14 The Question Analyzer 68

CHAPTER 3: RECATEGORIZING QUESTIONS BY INFERENTIAL ANALYSIS

3.0 Introduction 71
3.1 Contextual Inferences 72
    3.1.1 Conversational Scripts 74
    3.1.2 Generalized Inferences 77
        3.1.2.1 Single Word Questions 77
        3.1.2.2 Universal Set Inference 78
        3.1.2.3 Implicit Requests 79
            ATRANS Request Conversion 80
            MTRANS Request Conversion 81
            Performance Request Conversion . . 82
            Permission Request Conversion . . 83
            Function Request Conversion . . . 84
        3.1.2.4 Implicit Causality 85
            Enablement Conversion 85
    3.1.3 Conversational Continuity 86
        3.1.3.1 The Basic Rule 87
        3.1.3.2 The Last MLOC Update 88
        3.1.3.3 Pronominal Reference 88
        3.1.3.4 Focus-Based Continuity 90
        3.1.3.5 Knowledge-Based Continuity 93
3.2 Context-Independent Inferences 93
    Simple Request Conversion 94
    Frequency Specification Conversion 94
    Duration Specification Conversion 95
    Agent Request Conversion 96
3.3 Knowledge State Assessment Inferences 97
    Goal Orientation Conversion 98
    Specification Constraint Conversion 98
    Obvious Request Conversion 99

PREFACE TO MEMORY SEARCHES: FINDING AN ANSWER

0. Introduction 101
1. Deciding How to Answer a Question 101
2. Carrying Out Orders 103
3. Answering Better by Understanding More 103
4. Generation into English 106

CHAPTER 4: CONTENT SPECIFICATION

4.0 Introduction 109
4.1 Intentionality 111
  4.1.1 Mood 111
  4.1.2 Reliability 112
  4.1.3 Intentionality and Answers 112
4.2 Elaboration Options 113
    Verification Option 114
    Short Answer Option 115
    Single Word Option 116
    Correction/Explanation Option 116
    Inquiry/Explanation Option 118
    Request Explanation Option 119
    Delay Specification Option 120
    Condition Specification Option 120
    Inference Anticipation Option 121
    Mental State Description Option 122
  4.2.1 Preventative Inference Simulation 122
4.3 Category Trace Instructions 124
    Verifying Frequency Instructions 125
    Verifying Duration Instructions 126
  4.3.1 Category Trace Instructions vs. Elaboration Options 126
  4.3.2 Hardly Ever Is Never Very Often 127
4.4 Elaborating Elaboration Options 128

CHAPTER 5: SEARCHING MEMORY

5.0 Introduction 130
5.1 Causal Antecedent 131
  5.1.1 Script Structure Retrieval Heuristics . . 131
  5.1.2 Planning Structure Retrieval Heuristics . 133
  5.1.3 Causal Chain Retrieval Heuristics 134
    5.1.3.1 Non-Standard Inference Search . . 134
    5.1.3.2 Interference Search 135
    5.1.3.3 Script-Internal Goal Structures . 135
5.2 Goal Orientation 136
  5.2.1 Script Structure Retrieval Heuristics . . 136
  5.2.2 Planning Structure Retrieval Heuristics . 137
  5.2.3 Causal Chain Retrieval Heuristics 138
5.3 Enablement 138
  5.3.1 Script Structure Retrieval Heuristics . . 138
  5.3.2 Planning Structure Retrieval Heuristics . 139
  5.3.3 Causal Chain Retrieval Heuristics 140

5.4 Causal Consequent 140
    5.4.1 Script Structure Retrieval Heuristics . . 141
        5.4.1.1 Weird Event Search 141
        5.4.1.2 Interference/Resolution Search . . 141
        5.4.1.3 Main Act Search 142
    5.4.2 Causal Chain Retrieval Heuristics 142
        5.4.2.1 Default Path Departures 142
        5.4.2.2 Resolution Search 143
        5.4.2.3 Chronological Consequent Search . 143
5.5 Verification 144
5.6 Disjunctive 145
5.7 Instrumental/Procedural 146
    5.7.1 Script Structure Retrieval Heuristics . . 146
    5.7.2 Planning Structure Retrieval Heuristics . 146
    5.7.3 Causal Chain Retrieval Heuristics 147
5.8 Concept Completion 147
5.9 Expectational 148
5.10 Judgemental 148
5.11 Quantification 150
5.12 Feature Specification 150
5.13 Concluding Remarks on Retrieval Heuristics . . 151

CHAPTER 6: FOCUS ESTABLISHMENT

6.0 Introduction 152
6.1 Different Kinds of Focus 153
    6.1.1 Stress Intonation Patterns 153
    6.1.2 Syntactic Constructions 154
    6.1.3 Context and Focus 154
    6.1.4 World Knowledge and Focus 156
6.2 When Focus is Established 157
6.3 A Script-Based Focus Heuristic 158

CHAPTER 7: UNDERSTANDING WHAT DIDN'T HAPPEN

7.0 Introduction 166
7.1 Aroused Expectations 166
7.2 Violated Expectations 167
7.3 Answering Questions About Expectations 168
7.4 Answering Questions About Possibilities 169
7.5 Classification of Expectational Questions . . . 170
7.6 Script-Based Expectations 171
7.7 How to Remember Things You Forgot 172
    7.7.1 Ghost Path Generation 173
    7.7.2 Using Ghost Paths 177

CHAPTER 8: FINDING THE BEST ANSWER

8.0 Introduction 179
8.1 An Answer Selection Model 181
    8.1.1 Definitions 181
    8.1.2 More Definitions 183
    8.1.3 Selection Rules 184
    8.1.4 Implementing the Model 187
    8.1.5 ASP Output 188
8.2 Going Beyond Answer Selection 192
    8.2.1 What's Wrong with the Answer Selection Model 192
    8.2.2 A Retrieval Rule Incorporating MLOC Assessment 199
8.3 Looking Inside MBUILD's 207
    8.3.1 Plan-Based Retrieval Heuristics 210
    8.3.2 Retrieval Heuristics in PAM 211
8.4 Concluding Remarks 213

CHAPTER 9: CONCEPTUAL PRIMITIVES FOR PHYSICAL OBJECTS

9.0 Introduction 214
9.1 Object Primitives 216
    9.1.1 SETTING 217
    9.1.2 GESTALT 218
    9.1.3 RELATIONAL 219
    9.1.4 SOURCE and CONSUMER 220
    9.1.5 SEPARATOR and CONNECTOR 223
9.2 Applications for Object Primitives 224
    9.2.1 COIL 225
    9.2.2 State Descriptions 228
    9.2.3 Script Application 229
    9.2.4 Locational Specification 231
    9.2.5 Inference Mechanisms 233
        9.2.5.1 A Demon 233
        9.2.5.2 From Soup to Trains 236
9.3 Conclusions 243
    9.3.1 Theories of Human Memory 243
    9.3.2 Contextually Dynamic Memory 243
    9.3.3 Primitive Decomposition - Why Do It? . . . 245

CHAPTER 10: MORE PROBLEMS

10.0 Preface 248
10.1 Consistency Checks 248
10.2 Modeling Knowledge States 249
10.3 Conversation Theory 253
10.4 What Every Lawyer Should Know 255
10.5 General Q/A 256
10.6 Psychology and Artificial Intelligence 257

CHAPTER 11: PERSPECTIVE AND CONCLUSIONS

11.0 Preface 259
11.1 Other Q/A Systems 259
  11.1.1 Winograd (SHRDLU) 259
  11.1.2 Woods (LSNLIS) 261
  11.1.3 Waltz (PLANES) 262
  11.1.4 Scragg (LUIGI) 263
  11.1.5 Bobrow (GUS) 264
11.2 Summary 265

APPENDIX 1: THE PRIMITIVE ACTS OF CONCEPTUAL DEPENDENCY 267

APPENDIX 2: SCRIPTS & PLANS 271

APPENDIX 3: STORY REPRESENTATIONS 273

BIBLIOGRAPHY 279

CHAPTER 1

PROBLEMS, PREVIEWS AND PROGRAMS

1.0 Introduction

Question answering is a process. If we wish to program a computer to answer questions, we need some sense of what that process looks like. Human question answering is more than lexical manipulation; the cognitive mechanisms used in question answering operate on concepts underlying language. The processes of question answering must therefore be characterized as manipulations of conceptual information. This thesis presents a process model of question answering as a theory of conceptual information processing.

The difficulties involved in natural language question answering (Q/A) are not obvious. People are largely unconscious of the cognitive processes involved in answering a question, and are consequently insensitive to the complexities of answering questions. It is therefore necessary to become acquainted with the variety and scope of the difficulties involved; the problems intrinsic to question answering must be made visible. There is no way to consciously monitor how questions are understood, how memory is searched for an answer, or how that answer is expressed in language. A question is heard and an answer surfaces. It is very hard to appreciate the complexity of a process which is effortless and unconscious. But if we examine some examples of questions and answers, we can begin to realize how much is involved in the fundamental and 'simple' human ability to answer questions.

1.1 What's Hard About Answering Questions

Before anyone can answer a question they must understand what the question says. The processes which understand questions operate on various levels. To see how many kinds of interpretive processes are involved, we will look at some of the ways that questions can fail to make sense. From there we will go on to problems which are related to memory retrieval and the processes which operate on memory in order to form an answer.

1.1.1 Conceptual Analysis

Imagine you are walking down the street when a well dressed stranger comes up to you and says:

Q: Pardon me, but do you kronfid grodding slib?

Chances are you will not understand this question. Your failure to understand this question occurs at the lowest possible level - lexical processing. The first step of interpretive processing maps the lexical question into a conceptual structure; combinations of words are given meaning. This conceptual analysis is the only interpretive process involved in question answering which is language dependent. If I know only English I cannot understand questions in other languages. Once a question has been successfully mapped from its lexical expression to a conceptualization, the subsequent interpretive processes which interpret the question further are all language-independent processes. If I am learning Russian, I need to learn a vocabulary and grammar in order to understand questions which are stated in Russian, but I do not need to acquire new cognitive processes for answering questions once I know their conceptual meaning.

Principle #1:

The Memory Processes of Question Answering

are Independent of Language

The analytic mechanisms which map words to concepts when a question is understood are language-dependent mechanisms. Similarly, the generational mechanisms which express concepts in words are language-dependent. But the cognitive mechanisms of memory which make inferences about questions and search memory for an answer operate on a conceptual level which is independent of language.

1.1.2 Reference Recognition

In the course of obtaining a conceptual interpretation, various memory processes operate to see if all nominal references in the question are understood. For example, if I ask you:

Q: Which is bigger, a basenji or a komondor?

the question makes some sense conceptually, even if the words basenji and komondor are totally alien. The question can be understood to be asking for a size comparison between two things. But if the question refers to something which is completely unheard of, the question is not fully understood on the level of reference recognition.

Principle #2:

Questions are Understood on Many Different Levels

Full reference recognition does not guarantee that a question can be answered, but a lack of reference recognition does guarantee that a question can't be answered (at least knowledgeably). If it is known that basenjis and komondors are two breeds of dog, and nothing more is known about either breed, reference recognition will have been achieved, and the question will still be unanswerable. But if a conceptual reference for either basenjis or komondors is missing, the question can't possibly be answered.

1.1.3 Conceptual Categorization

When a question has been understood conceptually, it is ready to be classified conceptually. Conceptual categorization of a question determines which memory processes will be invoked to complete further interpretation. Suppose two people want to have dinner together but neither of them has any money, and there is no food at home. In this context, consider the following exchange:

Q: How are we going to eat tonight?
A: With silverware.

This answer indicates that the question failed to be interpreted correctly. It was placed in the wrong conceptual category. The answer indicates that the question was understood to be asking about the instrumentality involved in eating. But in the context given, the question should have been understood to be asking about the enabling conditions for eating. The question should be conceptually equivalent to asking, 'What are we going to do in order to eat tonight? - How are we going to solve this problem?'

Principle #3:

Understanding Questions Entails

Conceptual Categorization

In this thesis we will propose thirteen conceptual categories for questions. Two of these distinguish between questions that ask about enabling conditions and questions that ask about instrumentality. When a question is properly interpreted, it will be assigned to only one conceptual category. All answers to questions are produced by assuming one conceptual categorization. When a question is conceptually ambiguous (can be assigned to more than one conceptual category), the person being addressed must either clarify the question before attempting to answer it, or guess which category should be assigned and answer it on the basis of that assumed categorization. Conceptual categorization guides the subsequent processing of a question by dictating which interpretive mechanisms and memory retrieval strategies are to be executed.

1.1.4 Understanding in Context

The proper conceptual categorization of questions is dependent on the context in which the question occurs. Suppose I had been fixing dinner for you at my house, and just as I was about to set the table I remembered that I had loaned all my silverware to a friend. If I explain this to you and then ask:

Q: How are we going to eat tonight?

it would be quite reasonable for you to respond:

A: With our hands.

This answer is reasonable in the sense that it addresses the instrumentality of eating. Instrumentality is clearly what the question should be interpreted to be asking about in this context. But when the same question was asked in the context of having no food or money, an answer which responded to instrumental interpretation was inappropriate.

Principle #4:

Context Affects Conceptual Categorization

Processes that assign the proper conceptual categorization to a question must be sensitive to the context in which that question is asked. Questions cannot be correctly understood by processes which do not take into consideration contextual factors.

1.1.5 Conversational Interpretation

In human question answering dialogs, rules of conversational continuity are often invoked during the interpretative processing of questions:

John: Did Bill go to class?
Mary: No, he didn't.
John: Why not?
Mary: He was sick.

John: Did Bill go to class?
Mary: I don't know.
John: Why not?
Mary: I wasn't there.

In both these dialogs John asks Mary 'Why not?' In the first dialog he means 'Why didn't Bill go to class?' and in the second he means 'Why don't you know whether or not Bill went to class?' Rules of conversational continuity are needed in cases like these to complete a partial question correctly.

Principle #5:

Rules of Conversational Continuity

are Needed to Understand Some Questions

The mechanism which operates in cases like these is really very simple: When Mary has to interpret a question of John's in terms of previous dialog, she needs to know only the last concept which was communicated to John. In the first dialog Mary told John that Bill did not go to class. When John asks 'Why not?' she combines his partial question with the last concept communicated to him to get the question 'Why didn't Bill go to class?' In the second dialog, Mary told John that she doesn't know if Bill went to class or not. When John asks 'Why not?' she combines his partial question with the last concept communicated to get 'Why don't you know if Bill went to class?'

1.1.6 Inferences

Many questions rely on the listener's ability to infer something implicit in a question for the correct interpretation of that question:

Q: Do you drink?
A: Of course. All humans drink.

Q: Who wasn't in class yesterday?
A: George Washington and Moby Dick.

Q: Would you like to dance?
A: Sure. You know anyone who wants to?

While the answers given to these questions may be technically correct answers, they seem to have missed the point. One suspects that 'Do you drink?' is intended to mean 'Do you drink liquor?' and that 'Who wasn't in class yesterday?' must have meant 'Who wasn't in class yesterday who should have been there?' 'Would you like to dance?' should have been understood to mean 'Would you like to dance with me?' If these inferences are not made, answers which are literally correct but totally inappropriate will be given.

Principle #6:

A Good Answer Is More Than A Correct Answer:

Appropriateness Counts

When a question relies on the listener's ability to make inferences, missing information may derive from the conversational context of the question:

John: Hello Mary! Thanks for inviting me over.
Mary: I'm glad you could come. Sit down.
John: You certainly have a cheerful kitchen.
Mary: It was just painted. Some coffee?
John: No thanks, I don't drink coffee.

In this conversation 'Some coffee?' is an offer. Mary is asking John if he would like a cup of coffee. In order to interpret this question as an offer we must infer that Mary is in a setting where she has access to coffee (she isn't on a subway), Mary must have a relationship with John in which she treats him cordially (he is not a rapist) and they must be together (the question makes no sense over the phone).

John: Hello Mary! Thanks for inviting me over.
Mary: Would you like something to drink?
John: Some coffee?
Mary: Sure.

Now 'Some coffee?' is a polite response to an offer already made. John is saying he would like some coffee if she has any. Conversational context is a determining factor in the correct interpretation of these questions.

1.1.7 Knowledge-Based Interpretation

Very often we interpret questions correctly because we have knowledge about the world, things people do, and why they do the things they do. For example, suppose Mary hears that her friend John roller skated to McDonald's last night. She may very well ask:

Q: Why did John roller skate to McDonald's last night?

If someone answers her question:

A: Because he was hungry.

she will be justifiably impatient with that answer; she was not told what she wanted to know. Chances are she really wanted to know:

Q: Why did John roller skate instead of walk or drive or use some other reasonable means of transportation?

It was the act of roller skating that she was asking about, not the destination. A cooperative and reasonable respondent would have interpreted her question to be asking about the roller skating. But what is it that tells us which part of her question should be addressed? How do we know that the act of roller skating should receive attention and not the destination? This is a problem in knowledge-based focus establishment.

Principle #7:

Shifts in Interpretive Focus

Alter Meaning

Some questions are not fully understood until their focus has been determined. In many cases the focus of a question can be established only by knowing about how the world works and which things are more interesting than which other things. In the example just given the focus of our question is 'obvious' because adults frequently go to drive-in restaurants but they don't normally roller skate. For most adults, going roller skating is more unusual than going to McDonald's; it requires explanation. But if everyone knew that John was an eccentric health food nut who roller skates everywhere he goes, then the question:

Q: Why did John roller skate to McDonald's?

would be reasonably interpreted to be asking about McDonald's, not the roller skating. Our knowledge about John and what constitutes strange behavior for John would override our general knowledge about what people normally do and don't do.

Principle #8:

Focus Directs Attention to

Variations on Expectations

Interpretation of a question may force the listener's attention to favor some conceptual component of the question. Attention is generally drawn to those things which are unusual or which violate expectations.

In spoken dialogs, intonation patterns in speech very often establish the focus of a question. If McDonald's is stressed, it is clear that the speaker is interested in that particular component of the question. But in written questions, no such clues exist (unless we have italicized words). When no verbal (or visual) stress marks the focus of a question, heuristics must be invoked to determine which elements of the question are most deserving of attention. Attention naturally centers on the unexpected, neglecting those things which are predictable.

1.1.8 Memory-Based Interpretation

Sometimes a question does not have any conceptual components which are inherently deserving of focus on the basis of general world knowledge or specific knowledge of individual behavior. If I were to ask you:

Q: Why did John ask Mary to mail the card?

you can't establish the focus of this question by looking for what seems to be the most interesting thing in the question. The whole question looks pretty commonplace. People often ask each other to do things and there is nothing very exciting about mailing a card.

The focus of a question like this one relies solely on specific knowledge of the episode which the question refers to. Depending on the situation, this question could be answered:

A: Because Susan wasn't there.
A: Because he couldn't afford a personal delivery service.
A: Because he had no right to order her.

The answer 'Because Susan wasn't there,' could be a response to two possible interpretations of the question. The question (Why did John ask Mary to mail the card?) could be asking about either John as the actor, or Mary as the recipient of John's request. The answer could therefore mean that either John asked Mary because Susan wasn't there to ask Mary, or, John asked Mary because Susan wasn't there to be asked. If the question occurred in a real context (say in reference to a story) there would be no ambiguity to the answer since only one would make sense in context. 'Because he couldn't afford a personal delivery service,' is an answer which can be produced only if the question is interpreted to be asking about the act of mailing. And 'Because he had no right to order her,' comes about by interpreting the question to be asking about the act of asking. It would be easy to make up stories for which each of these three answers makes sense.

Principle #9:

Ambiguity of Focus

Rarely Occurs in Context

When a question like this one is answered in the context of a story, focus is immediately established on the basis of what's in memory. If we heard a story where John was going to ask Susan to mail a card for him, but he ended up asking Mary because Susan was busy, then the question will immediately be understood to be asking about Mary vs. Susan.' Other possible interpretations which would have resulted from different focus establishment are never considered.

1.1.9 Selecting the Best Answer

Once a question has been fully interpreted it is time to look for an answer. Memory searches must be conducted which are guided by the conceptual category of the question at hand. An immediate problem arises when the memory search turns up more than one reasonable answer to a question. For example, consider the following story:

> John took a bus to New York. Then he took a subway to Leone's. But on the subway his pocket was picked. He went into Leone's and had some lasagna. When the check came, he realized he couldn't pay the check. He had to wash dishes.

If asked:

Q: Why did John wash dishes at Leone's?

there are many reasonable answers to this question:

A: Because he couldn't pay the check.
A: Because he had no money.
A: Because he was pickpocketed.

All of these answers are perfectly correct, but the last one seems to be the best answer in terms of information conveyed. Given that John was pickpocketed one can infer that he had no money and therefore couldn't pay the check. But if told that he couldn't pay the check, or he had no money, there is still a missing causality: why couldn't he pay the check? why didn't he have any money? In fact the first two answers are poor answers because not being able to pay a check can be inferred from washing dishes in a restaurant, and not having (enough) money can be inferred from not being able to pay the check. An answer which provides the questioner with new information is preferable to an answer which tells the questioner something he could have inferred for himself.

Principle #10:

The More Inferences an Answer Carries,

the Better the Answer

When the memory search finds an answer, it must take into account what the questioner can be expected to know on the basis of the question asked. The total information conveyed by an answer is not contained by the explicit answer alone. Inferences made by the questioner upon hearing the answer augment the explicit answer. An answer must be chosen on the basis of the inferences it carries.

1.1.10 State Descriptions

Suppose John is eating in a dining car and when the train starts up, the soup spills. Something very interesting happens when people answer the question:

Q: Where was the soup?

In an informal experiment, answers to this question tended to say one of two things: in a bowl or on the table. This is very interesting when considered in view of everything people know about the setting of a dining car and how people eat in dining cars. People can infer that the soup was in a bowl, the bowl was on a plate, the plate probably rested on a placemat or a table cloth which was in turn on a table, and the table was on the floor of the dining car which was part of the train. So with all of this information, why are the bowl and the table singled out to describe the location of the soup?

Some people will answer the question 'In the dining car.' But the answer 'On the train.' is odd, and it is very hard to imagine anyone answering 'On a plate.' There must be some rules of locational specification which are used to single out salient references. But what do these rules look like? Suppose there were a rule which said to specify the immediately contingent object of containment or support. This rule would account for the response:

Q: Where was the soup?
A: In a bowl.

But it wouldn't account for:

Q: Where was the soup?
A: On the table.

What rule accounts for this answer? It is not enough to specify the closest supporting object because that rule would result in:

Q: Where was the soup?
A: On a plate.

The conceptual rules which guide locational specification must be general enough to work in a variety of settings. In the train example it looks like it might be pretty safe to specify the immediately contingent object of containment or location since this rule yields one acceptable answer:

Q: Where was the soup?
A: In a bowl.

But suppose John went into the kitchen and poured himself some milk. Now answer the question:

Q: Where did John get the milk?

A natural answer to this question is 'From the refrigerator.' But no one imagines the milk to be sitting in a puddle at the bottom of the refrigerator. There is a milk container of some sort which is the contingent object of containment. Yet very few people will mention a milk container when describing where the milk was. So a rule which specifies the object of immediate containment will not produce natural answers in all contexts.

In order to answer questions like these as a person would, we must have conceptual representations for objects which tell us that the link between milk and milk containers is somehow too obvious to mention while the link between soup and bowls is not. The relational notions of containment and support do not seem to inspire rules of retrieval which are both simple and effective in terms of producing good answers. A conceptual representational system for physical objects must be utilized in the development of simple and effective retrieval rules for locational specification. The foundations for such a representational system will be proposed in this thesis.

Principle #11:

A Difficult Retrieval Problem May Point to

Weaknesses in the Memory Representation

Sometimes questions which ask for locational specification must be answered according to who is asking the question and why:

Q: Where is Deerfield, Illinois?
A: 87 54' long. 42 12' lat.
A: Near Lake Michigan, about 30 miles north of Chicago.
A: Next to Highland Park.
A: On the planet Earth.

Each of the above answers is correct, but in any given context some of these answers will be useless. If John lives in Illinois and his neighbor asks him where Deerfield is, he will probably not appreciate an answer specifying the longitudinal and latitudinal descriptors. Chances are he would find 'On the planet Earth,' to be an insulting answer. Yet there are contexts in which each of these answers would be appropriate. An airline pilot might prefer the first answer, and a non-earthling character in a novel might find the last answer to be the most meaningful. When people answer questions they must often assess the knowledge state of the person they are addressing. 'Next to Highland Park,' is a good answer only if you know where Highland Park is. 'Near Lake Michigan, about 30 miles north of Chicago,' is liable to satisfy anyone who has a rough familiarity with the U.S.A. '42 12' lat. 87 54' long,' is a good answer only when a technical global specification is required. 'On the planet Earth,' is a poor answer to anyone who lives on earth.

Principle #12:

Good Answers Can Involve

Knowledge State Assessment

This problem of finding a suitable description when answering locational specification questions was described by Donald Norman [Norman 1972] as the Empire State Building problem (Where is the Empire State Building?). Locational specification questions illustrate very clearly how answers to questions must strive to fill in the gaps of the questioner's knowledge state.

1.1.11 Content Specification

Somewhere in the question answering process a decision must be made about how much information is going to be put into the answer. For that matter, the system must be able to decide if the question is going to be answered honestly, deceptively, sarcastically, or in any other of a hundred different ways. For example, suppose questions are asked about the following story:

> John went to a restaurant and ordered a hot dog.
> But the waitress said they didn't have any so he
> had a hamburger instead.

If asked:

Q: Did John eat a hot dog?

there are at least three ways that this question could be reasonably answered:

A: No.
A: No, John ate a hamburger.
A: No, John ate a hamburger because there were no hot dogs.

Each of these answers is correct but each one provides a different amount of information. Some part of the question answering process must be sensitive to factors which control the system's disposition in terms of reliabilty and relative information content. There must be mechanisms which control memory retrieval processes and provide instructions about how an answer is to be constructed.

Principle #13:

The Same Question Doesn't Always Get the Same Answer

A question answering system must be able to instruct its retrieval mechanisms how to form answers so that the system does not always return the same answer to the same question. Answers must vary appropriately in response to contextual and motivational factors.

1.1.12 Accessing Memory

When a story is read, a conceptual structure representing the story is generated in memory which encodes events mentioned in the story and inferences made by the reader at the time of understanding. [Schank 1974b, 1975b]. This structure is language-independent, existing as a purely conceptual record of the story. When answering questions about a story this story representation is accessed by memory searches. Question answering is therefore concerned with the form of information in memory and the processes which access that information. In designing memory representations for stories, some assumptions must be made about what belongs in a story representation. When a story describes a chronological sequence of events, it is reasonable to assume that the events preserved in the story representation should encode only information about what happened, where something that happened is either an event mentioned explicitly in the story or one which was inferred by the reader. For example, if a story says that John ate a hamburger at a restaurant, the memory representation should contain the inference that John ordered a hamburger, but it shouldn't contain negative information like the fact that John didn't order a pizza.

When answering questions about a story, the story representation which was generated at the time of understanding is examined by memory searches. But there are times when the story representation by itself is not adequate for finding an answer and more inferences must be made at the time of question answering. For example, if we adhere to the assumption that a story representation should only contain information about what happened (and what we infer must have happened), then how can we go about answering questions which ask about things that didn't happen?

> John went to a restaurant and ordered a hamburger. But the waitress said they didn't have any so John left.

If asked:

> Q: Why didn't John eat a hamburger?

there is nothing in our story representation which says anything about eating a hamburger, or eating anything, for that matter. The story representation simply records a chain of events which includes John going into the restaurant, sitting down, the waitress coming over, John telling her he wants a hamburger, the waitress telling him there were none, and John getting up and leaving. Yet we should be able to answer this question with a response like, 'Because they didn't have any.'

> Principle #14:
>
> Sometimes Inferences Must Be Made
>
> at the Time of Question Answering

Some additional inference mechanisms must be invoked in order to produce this answer. If, in the context of this story, we had been asked 'Why didn't John swim across the lake? the question wouldn't make sense. The inference mechanisms which are invoked to answer these questions are capable of seeing when a why-not question makes sense by determining if the act in question would have occurred had the story taken a different turn. 'Why didn't John eat a hamburger?' makes sense because we expected him to eat a hamburger after he ordered, until the waitress told him there were none. But 'Why didn't John swim across the lake?' makes no sense in this story since we never expected John to swim across a lake.

1.1.13 Constructing Information

People are able to answer many questions by deriving information which is not explicitly present in memory for straightforward retrieval.

> Q: Who was President when you were entering the sixth grade?
> Q: What is 56 times 8?
> Q: How far is New York from Boston?

Each of these questions can be answered, but probably not immediately. The first question may require an associative path from the sixth grade, to your age at the time of the sixth grade, to the year, to the President. Or perhaps there was some event concerning the (then) current President which you know occurred at about the time you entered the sixth grade, and so you can answer the question by remembering that event. The multiplication question is liable to require an arithmetic calculation. And the distance question may be a matter of simple retrieval if you happen to 'know' that fact. Otherwise the answer could be derived by knowing how long it takes to drive or fly between to Boston and New York.

Principle #15:

You Can't Expect to Always Find

Exactly What You Had in Mind

People seem to be very flexible in memory retrieval tasks. When faced with a question like

Q: How many days does July have?

many people resort to a little rhyme: '30 days hath September, April, June, and November, ...' But if you ask someone this question on the 31st of July, there is a good chance that they will be able to answer without resorting to the rhyme if they are aware of the date. People tend to know that no month has more than 31 days, so if a month has at least 31 days, it must have exactly 31 days. Given that July has at least 31 days, this little bit of reasoning is faster than running through the rhyme.

1.1.14 Finding Only What's Needed

Efficient retrieval entails an ability to recognize when enough information has been seen in order to answer the question. Suppose you have read the following:

> John boarded the train in Boston Monday morning. He slept most of that afternoon and had dinner in the dining car. Then he played cards in the club car and finally went to bed around 2:00. The train got into Chicago after lunch the next day.

Now if asked:

Q: How long was John on the train?

you will try to piece together as accurate an answer as possible. Running back over the time line of the story you can piece together his boarding in the morning, the overnight, and the train's arrival after lunch in order to arrive at his being on the train a little over a day. But suppose instead you had been asked:

Q: Did John spend a week on the train?

You can answer 'no' to this question much faster than you could answer the previous question. This answer is easily derived from the fact that John spent only one night on the train. No further information from your memory of the story is needed to answer the question. More information had to be examined to answer the first question than was needed for the second.

Principle #16:

A Good Search Strategy Knows When It Has the Answer:

Smart Heuristics Know When to Quit

1.2 Where We're Going

The scope of problems involved in designing a question answering system ranges in a number of directions and covers a tremendous amount of territory. Section 1.1 does not constitute a definitive survey of problems; it was intended only to convey some sense of the issues we will address.

There are many problems in designing a question answering system. These problems do not exist in isolation of one another; solutions in one area often contribute to solutions elsewhere. It is therefore more productive to view the issues from a vantage point which provides a global sense of the entire question answering process. It would be very difficult to devote a chapter to each problem and still describe a cohesive picture of question answering. For this reason, a process model for question answering is described by following a question from the initial understanding of words in the question through to the final lexical expression of an answer. After this fundamental process model is presented, specific areas of difficulty are identified and discussed in detail.

Within this organization, the problems of the preceding section will be distributed throughout as follows:

Preface to Interpretation: Understanding Questions

> The chapter is an overview which introduces the processes underlying Principles #1-6. An outline of the interpretive process model is given which describes how the issues of conceptual analysis, reference recognition, conceptual categories, understanding in context, conversational interpretation, and inference, all fit together in the processing of a question.

Chapter 2: Conceptual Categories for Questions

Chapter 2 elaborates Principle #3 and discusses conceptual categorization. Thirteen conceptual categories are presented and the analysis mechanism which assigns a category to a question is described.

Chapter 3: Recategorizing Questions by Inferential Analysis

Chapter 3 is concerned with Principles #4, 5, and 6. Understanding in context, conversational interpretation, and inferences are all topics in inferential analysis.

Preface to Memory Searches: Finding an answer

This chapter is an overview of the memory search which introduces the processes underlying Principles #7, 8, 9, and 13. Content specification and the memory search strategies needed to produce an answer are outlined here.

Chapter 4: Content Specification

Chapter 4 elaborates Principle #13 and explains how content specification controls the amount of information which goes into an answer.

Chapter 5: Searching Memory

Chapter 5 describes search heuristics for finding an answer.

Chapter 6: Focus Establishment

Chapter 6 is concerned with Principles #7, 8, and 9. Knowledge-based interpretation and memory-based interpretation is explored here.

Chapter 7: Understanding What Didn't Happen

Chapter 7 presents an example of Principle #14 by describing a technique for accessing memory.

Chapter 8: Finding the Best Answer

Chapter 8 explores Principles #10 and 12. This chapter is primarily concerned with selecting the best answer to a question.

Chapter 9: Conceptual Knowledge of Physical Objects

Chapter 9 illustrates Principle #11 while exploring problems in state descriptions. A system for representing physical objects by decomposition into primitives is proposed.

1.2.1 How to Survive the Trip

Whenever one confronts a nine course dinner there is always the danger of getting glutted before dessert or maybe even before the main courses. It sometimes helps to know what you're up against before you start.

It is assumed that the reader is familiar with Conceptual Dependency Theory. It would be sufficient to have read Chapter Three of Conceptual Information Processing [Schank 1975a] or Chapter Five of Computer Models of Thought and Language [Schank & Colby 1973]. Appendix 1 reviews the primitive acts and causal chain theory for those who feel rusty.

It is also assumed that the reader is somewhat familiar with the theory of scripts and plans and their use in story understanding. Anyone unfamiliar with scripts and plans should consult Scripts, Plans, Goals, and Understanding [Schank & Abelson '77]. Other introductory references include: [Schank & Abelson 1975, Schank & the Yale AI Project 1975, Cullingford 1975, Wilensky 1976, Lehnert 1977]. Appendix 2 reviews scripts and plans and summarizes the script-related terminology used in this thesis.

If the reader wants a minimal overview of the theory, it would be sufficient to read the two prefatory chapters. These prefaces serve as introductions to the chapters which describe the basic process model. The preface to interpretation (understanding questions) is between Chapters One and Two. The preface to memory searches (finding an answer) is between Chapters Three and Four.

The basic process model is described in Chapters Two through Five. A reader who wants only a basic introduction to the problem could stop after Chapter Five. But anyone interested in the real difficulties will find the more challenging issues discussed in Chapters Six through Ten.

1.3 QUALM

QUALM is a computational model of question answering. As such, there is a computer program which is an implementation of QUALM. This program runs in conjunction with two larger systems, SAM and PAM. Both SAM and PAM are comprehensive story understanding systems which input stories in English and generate internal memory representations for what they read. These story representations can then be accessed by processes which are designed to paraphrase, summarize, and answer questions about the stories read. QUALM is responsible for the question answering capacities of SAM and PAM. Both systems are modularized so that parsers and generators can be attached for different input and output languages. At the moment SAM and PAM operate with English input and produce paraphrase output in English, Spanish, Russian, Dutch, or Chinese. All of the question answering done by SAM and PAM has been in English. QUALM itself is language independent. No changes would have to be made to QUALM in order for SAM and PAM to understand and answer questions in different languages.

Two experimental programs have been designed in addition to QUALM. These are ASP (Answer Selection Program) and COIL (Conceptual Objects for Inferencing in Language). These were both designed to explore specific issues which were relevent to the design of QUALM as a system independent of SAM and PAM.

ASP is a small interactive system which inputs a question about a story along with a set of possible answers. It asks the user to characterize those answers according to instructions given by the program. ASP then picks out what it considers to be the best answer to the question.

COIL is a larger system designed to generate memory representations for stories and answer a small variety of questions about the stories on the basis of its memory representation. COIL incorporates a toy parser and generator. Problems in parsing and generation are peripheral to the theoretical issues addressed by COIL. COIL was primarily designed to see how a representational system for physical objects could be exploited by inference and retrieval mechanisms.

QUALM is an independent system which can be attached to any story understanding system generating story representations according to the theories of predictive understanding by scripts and plans. Although the computer implementation of QUALM is limited to the context of answering questions about stories, the theoretical model goes beyond this particular context. As a theoretical model QUALM describes general question answering, where question answering in its most general form is viewed as a verbal communication device between people.

1.4 Computer Programs Using QUALM

The process model for question answering which is described in this thesis is currently implemented in two computer programs, SAM and PAM. These two programs have been designed to understand stories and answer questions about what they have read. In the following pages are examples of stories which have been processed by these systems along with questions and answers. The stories and questions were input in English as they appear here, and the answers given here are the actual English-generated responses which were returned by the computer.

In all cases, the system first reads the story one sentence at a time and generates a memory representation for that story as it reads. When the story has been completed, questions are asked. Each question is processed and answered before going on to the next one.

The last two programs, the Answer Selection program and COIL, are experimental systems which were designed to explore issues raised by QUALM.

1.4.1 SAM

SAM was developed by the Artificial Intelligence Project at Yale. It processed its first story in 1975 and has been undergoing continual expansion ever since. The actual programming of SAM has relied on the persistent efforts of many: Richard Cullingford, Gerald DeJong, Leila Habib, James Meehan, Richard Proudfoot, Christopher Riesbeck, Walter Stutzman, and Robert Wilensky.

The following stories and question answering dialogs represent actual input and output. The stories and questions were read by the system just as they appear here. The answers are the exact English outputs generated by SAM.

****

John went to a restaurant. The hostess seated John. The hostess gave John a menu. The waiter came to the table. John ordered a lobster. He was served quickly. He left a large tip. He left the restaurant.

Why did John go to the restaurant?

SO JOHN COULD EAT A MEAL.

Did John go to a table?

JOHN WENT TO A TABLE.

Why did John go to a table?

SO JOHN COULD SIT DOWN IN A CHAIR.

Did the waiter give John a menu?

JOHN GOT A MENU FROM THE HOSTESS.

Who gave John a menu?

JOHN GOT A MENU FROM THE HOSTESS.

Why did the hostess give John a menu?

SO JOHN COULD ASK THE WAITER FOR A MEAL.

What happened when the hostess gave John a menu?

JOHN READ THE MENU, THE WAITER SAW THAT JOHN WAS AT THE TABLE, THE WAITER WENT TO THE TABLE.

Did John enjoy the lobster?

I DON'T KNOW.

Did the waiter give John the check?

JOHN GOT THE CHECK FROM THE WAITER.

Why did John leave the waiter a large tip?

PROBABLY JOHN REMEMBERED THE WAITER SERVED JOHN A LOBSTER QUICKLY.

Did John pay the bill?

JOHN PAID THE CHECK.

Why did John pay the check?

SO JOHN COULD LEAVE THE RESTAURANT.

****

John went to a restaurant. He sat down. He got mad. He left.

Did a waiter come to the table?

NO.

Did John order?

NO.

****

John went to New York by bus. On the bus he talked to an old lady. When he left the bus, he thanked the driver. He took the subway to Leone's. On the subway his pocket was picked. He got off the train and entered Leone's. He had some lasagna. When the check came, he discovered he couldn't pay. The management told him he would have to wash dishes. When he left, he caught a bus to New Haven.

Who went to New York?

JOHN WENT TO NEW YORK.

Why did John go to New York?

BECAUSE JOHN WANTED TO GO TO LEONE'S.

Why did John go to Leone's?

BECAUSE JOHN WANTED TO HAVE SOME LASAGNA.

How did John get to New York?

JOHN TOOK A BUS TO NEW YORK.

Did a thief pick John's pocket?

A THIEF PICKED JOHN'S POCKET.

Did John order lasagna?

JOHN ORDERED LASAGNA.

Did John eat a steak?

NO, JOHN ATE LASAGNA.

What did John eat?

JOHN ATE LASAGNA.

Did John order lobster?

NO, JOHN ORDERED LASAGNA.

Did John eat lasagna?

JOHN ATE LASAGNA.

Did John pay the check?

NO, JOHN DIDN'T HAVE ANY MONEY.

Did John leave the waitress a tip?

NO, JOHN DIDN'T HAVE ANY MONEY.

Did John leave Leone's?

JOHN LEFT LEONE'S.

Why did John take the subway from Leone's?

SO JOHN COULD TAKE A BUS TO NEW HAVEN.

How could John take the subway from Leone's?

PROBABLY JOHN HAD A TOKEN.

How did John get to Leone's?

JOHN TOOK A BUS TO NEW YORK AND THEN HE TOOK THE SUBWAY TO LEONE'S.

Did anything unusual happen on the bus?

I DON'T REMEMBER ANYTHING.

Did anything unusual happen on the subway?

A THIEF PICKED JOHN'S POCKET.

What did John order?

JOHN ORDERED LASAGNA.

Who gave John a menu?

THE WAITRESS GAVE JOHN A MENU.

Did anything unusual happen at the restaurant?

JOHN DISCOVERED THAT HE COULDN'T PAY THE CHECK AND SO HE HAD TO WASH DISHES.

Why couldn't John pay the check?

BECAUSE JOHN DID NOT HAVE ANY MONEY.

Why didn't John have any money?

BECAUSE A THIEF PICKED JOHN'S POCKET.

How did John get back to New Haven?

JOHN WENT FROM LEONE'S BY SUBWAY AND THEN HE WENT TO NEW HAVEN BY BUS.

How was John able to take the bus to New Haven?

PROBABLY JOHN HAD A TICKET.

****

John went to a restaurant and the hostess gave him a menu. When he ordered a hot dog the waitress said that they didn't have any. So John ordered a hamburger instead. But when the hamburger came, it was so burnt that John left.

Did the waitress give John a menu?

NO, THE HOSTESS GAVE JOHN A MENU.

Did John order a hot dog?

YES.

Did John eat a hot dog?

NO, THE WAITRESS TOLD JOHN THEY DIDN'T HAVE ANY HOT DOGS.

What did the waitress serve John?

THE WAITRESS SERVED JOHN A HAMBURGER.

Why didn't John eat the hamburger?

BECAUSE THE HAMBURGER WAS BURNT.

Was John angry?

PROBABLY.

Why didn't John pay the check?

JOHN WAS ANGRY BECAUSE THE HAMBURGER WAS BURNT AND SO HE LEFT.

Why didn't John eat a hot dog?

BECAUSE THE WAITRESS TOLD JOHN THEY DID NOT HAVE ANY HOT DOGS.

What happened when John ordered the hamburger?

THE WAITRESS GAVE THE ORDER TO THE COOK, THE COOK PREPARED THE HAMBURGER, THE COOK GAVE THE HAMBURGER TO THE WAITRESS, THE WAITRESS SERVED JOHN THE HAMBURGER.

What happened when John ordered the hot dog?

THE WAITRESS TOLD JOHN THEY DIDN'T HAVE ANY HOT DOGS.

****

Friday evening a car swerved off Route 69. The vehicle struck a tree. The passenger, a New Jersey man, was killed. David Hall, 27, was pronounced dead at the scene by Dr Dana Blauchard, medical examiner. Frank Miller, 32, of 593 Foxon Rd, the driver, was taken to Milford Hospital by Flanagan Ambulance. He was treated and released. The passenger was extricated from the vehicle by the Branford Fire Department. No charges were made. Patrolman Robert Onofrio investigated the accident.

Was anyone killed?

YES, DAVID HALL DIED.

Was anyone hurt?

YES, FRANK MILLER WAS SLIGHTLY INJURED.

Why was Frank Miller hurt?

BECAUSE THE AUTOMOBILE HIT THE TREE.

Did Miller go to the hospital?

YES, HE WAS IN THE MILFORD HOSPITAL.

How did Miller got to the hospital?

THE FLANAGAN AMBULANCE COMPANY TOOK HIM TO THE MILFORD HOSPITAL.

1.4.2 PAM

PAM was developed by the Yale Artificial Intelligence Project in 1976. The actual programming of PAM has been primarily the work of one person: Robert Wilensky. Supporting programs for PAM were handled by Gerald DeJong, Christopher Riesbeck, and Walter Stutzman.

The following stories and question answering dialogs represent verbatim input and output.

****

John loved Mary but she didn't want to marry him. One day, a dragon stole Mary from the castle. John got on top of his horse and killed the dragon. Mary agreed to marry him. They lived happily ever after.

Why did John get on his horse?

BECAUSE HE WANTED TO BE NEAR MARY.

Why did Mary agree to marry John?

BECAUSE SHE WAS INDEBTED TO HIM.

Why did John kill the dragon?

BECAUSE HE WANTED MARY TO NOT DIE.

****

John disliked his job as the bank janitor. One day, a thief tried to rob the bank. John jumped on the thief and knocked him out. The bank president made John the bank guard. John enjoyed his new job.

Why did John knock the thief out?

BECAUSE HE WANTED THE THIEF NOT TO HAVE THE BANK'S MONEY.

Why did the bank president make John the bank guard?

BECAUSE THE BANK WAS INDEBTED TO JOHN.

1.4.3 ASP

ASP is a small program which was designed in 1975 to investigate theories concerned with finding the best answer to a question. Its implementation required no efforts outside of my own. ASP is given a set of answers to a question about a story and ASP tries to select the best answer. ASP makes its decision by asking the user to respond yes or no to a sequence of questions about the answers provided. ASP has no memory representation for the story and no understanding of the question or the answers given. It chooses an answer on the basis of selection rules which respond to the yes or no answers of the user.

The following dialog represents an interactive session with ASP. ASP has the answers it was given stored in memory as lists so it can ask questions about pairs of answers by combining those questions with fixed connectives.

****

Story Context:

One morning John noticed that his dog was having trouble walking. That afternoon he took it to the vet.

Q: Why did John take his dog to the vet?

A1: It was sick or injured.

A2: It was having trouble walking.

A3: He wanted to make it well.

The following dialog represents an interactive session with ASP. For this execution ASP has been given the lexical answers A1, A2, and A3, as initial input to the program.

```
              ***** Interactive Session Begins *****


ASP:  GIVEN THE CONTEXT OF THE STORY, CAN EITHER

           He wanted to make it well

           OR

           It was sick or injured

           BE INFERRED FROM ONE ANOTHER?

User:  YES
```

ASP: DOES IT MAKE SENSE TO SAY

He wanted to make it well

BECAUSE

It was having trouble walking?

User: YES

ASP: DOES IT MAKE SENSE TO SAY

He wanted to make it well

AND HE KNEW THAT

It was having trouble walking?

User: YES

ASP: THE BEST ANSWER IS BECAUSE

It was having trouble walking

1.4.4 COIL

COIL was developed at the Xerox Palo Alto Research Center during the summer of 1976. It was written in an experimental implementation of KRL (Knowledge Representation Language) which was developed by Daniel Bobrow and Terry Winograd [Bobrow & Winograd 1977].

Unlike SAM and PAM which answered all of their questions after reading an entire story, COIL interrupts the input story to answer questions.

****

John picked up a newspaper. He went from the hall into the kitchen and got some milk.

Where did John come from?

THE HALL.

Where did the milk come from?

THE REFRIGERATOR.

But the milkcarton was empty so he threw it out.

Where did the milkcarton go to?

THE GARBAGE BAG.

Why did John throw the milkcarton into the garbage bag?

BECAUSE THE MILKCARTON WAS EMPTY.

He turned on the light and radio. Then he listened to music and read.

Where did the music come from?

THE RADIO.

What did John read?

THE NEWSPAPER.

Why did John turn on the light?

SO HE COULD READ THE NEWSPAPER.

Why did John turn on the radio?

SO HE COULD HEAR THE MUSIC.

Preface to Interpretation: Understanding Questions

0. Introduction

QUALM can be split up into roughly two fundamental processes: understanding the question and finding an answer. In Chapters Two and Three we will describe those processes of QUALM which are devoted to understanding questions. Questions are understood on different levels, and complete interpretation on preliminary levels must take place before the more comprehensive interpretation of subsequent levels can be attempted.

1. Four Levels of Understanding

All questions must pass through four levels of interpretive analysis before a memory search can begin to look for an answer. Interpretation on one level must be completed before interpretation on the next level can begin. The four successive levels are:

(1) Conceptual Parse
(2) Memory Internalization
(3) Conceptual Categorization
(4) Inferential Analysis

The first interpretive process, parsing, is the only one which is language dependent. Internalization, Conceptual Categorization, and Inferential Analysis all operate within Conceptual Dependency, a language-independent meaning representation [Schank 1975a]. This means that questions in any language can be processed by changing only the parser: English questions require an English parser, Russian questions require a Russian parser, and so forth. But the other interpretive modules which perform Internalization, Conceptual Categorization, and Inferential Analysis, do not have to be changed to accommodate questions in different languages. If an English parser were replaced by a Russian parser, the other interpretive modules would require no adjustment in order to function with the new parser.

Of the four interpretive processes, only the last two, Conceptual Categorization and Inferential Analysis are concerned solely with questions. The Conceptual Parser and Internalization programs are general language processing mechanisms, designed to deal with declarative statements as well as questions. These first two levels of interpretation are actually a front end for QUALM; they are not, strictly speaking, a part of QUALM. In fact, these analysis programs were independently developed long before anyone thought about applying them to the task of question answering. Christopher Riesbeck has been refining his Conceptual Parser (ELI) since its original implementation in 1974 [Riesbeck 1975, Riesbeck & Schank 1976]. Versions of the Internalization program (MEMTOK, TOK) have been operating since 1975 as a part of the SAM system [Cullingford 1977].

To understand the interpretive processes within QUALM, what they do and how they do it, it is not necessary to know how the Parser and Internalization programs work. But for the sake of an overall picture, we will take a little time to explain what they do (and don't do) so the reader will have some sense of the entire understanding process which begins with a string of words ending in a question mark.

2. The Conceptual Parse

When a question passes through the parser it is translated into a conceptual meaning representation, Conceptual Dependency. Once the Conceptual Dependency representation for a question has been produced, that conceptual representation is all the system needs to work with from that point on. None of the processes which follow require knowledge of what words or syntactic constructions were in the original lexical question.

The Conceptual Dependency representation of a question generated by the parser constitutes one level of question interpretation. Understanding of the question on this level is generally a literal or naive understanding. For example, the parser will understand:

Q1: Can you tell me where John is?

to mean 'Are you capable of telling me where John is?' If the interpretation ended here we would have no way of knowing that this is probably a request for information about John's location. The person asking Q1 wants to know where John is. He will probably not be satisfied to find out that the one he is addressing is merely capable of saying where John is. But the parser cannot know that this question is a request for information. As far as the parser is concerned, Q1 is a simple inquiry about whether or not the listener is capable of telling the questioner the whereabouts of John. Higher memory processes within the interpreter must be summoned to complete the interpretation of this question in order to arrive at the intended meaning of Q1.

This does not mean that the parser is operating in isolation of general world knowledge, context, or higher memory processes. On the contrary, parsing strategies often interact with other memory processes in order to arrive at the correct interpretation of verbs and nouns [Riesbeck & Schank 1976]. For example, if the parser is processing the sentence 'John walked into a restaurant and ordered a hamburger,' the parser will receive from the Script Applier a preferred word sense for the verb 'to order' which predicts that 'ordered' in this sentence means to communicate a request for a meal (instead of issuing a command or establishing a sequential arrangement of some sort). This preferred sense of the verb is established before the parser even gets to the object of the verb, the hamburger.

The parser interacts with contextual memory processes primarily for the sake of anticipating correct word senses. Suppose two people meet on the street and one asks the other:

Q2: Do you have a light?

The parser would understand that this question is equivalent to asking:

> Q3: Do you have in your immediate possession an object which is capable of producing a flame?

If Q2 had been asked in the context of two car mechanics working on a car engine in a garage, the parser would be capable of understanding that Q2 is equivalent to asking:

> Q4: Do you have in your immediate possession an object which is capable of producing light?

The parser would pick up a different sense of the word 'light' from a car mechanic script. It still doesn't understand the request aspect of the question, but it does receive word sense priorities from script applier processes which keep track of currently active scripts. The parser is sensitive to context in its conceptual analysis of statements and questions. But it is not capable of total understanding.

Complete understanding of Q2 recognizes this question to be a request for a lit match or lighter. This understanding is accompanied by a strong expectation that the questioner wants to light a cigarette. Such a level of predictive understanding can only be achieved by processes which have knowledge about questions and why people ask them. The parser does not have access to such knowledge. A correct parse of a question signifies the first level of understanding on which all subsequent interpretations are based. The proper conceptual parse of a question may not represent the intended meaning of that question; but when the parser's interpretation is not correct, it is better described as an incomplete interpretation rather than a wrong interpretation.

3. Memory Internalization

All sentences and questions must pass from an initial parse to an internalized parse. An internalized parse is a rewriting of the initial parse which substitutes pointers to memory tokens for all nominal references in the conceptualization. For example, the initial parse of the sentence 'John went to a restaurant,' will place in the Actor slot the word JOHN. When this conceptualization is internalized, memory is consulted to find out if previous processing has already established a memory token for a human with the name John. If so, JOHN is replaced by a pointer to the memory token already established (say GN001). If no previous reference to someone called John has occurred, memory will find no token for John and one will have to be created. In either case, the word JOHN will be replaced with a pointer to a memory token. When all nominal references in the initial parse have been replaced with pointers to memory tokens, we then have an internalized parse of the sentence.

The Internalization process establishes nominal referents and integrates new information into memory token descriptions of objects and people. For example, newspaper stories often begin like:

> John Doe of 4616 Lakewood Drive, Bingham, a construction worker, was killed yesterday when ...

The Internalization program is responsible for creating a memory token and organizing properties within it:

```
G0001:
    ISA = Human
    First Name = John
    Last Name  = Doe
    Sex = Male
    Age = >18 yrs.
    Occupation = Construction Worker
    Street Address = 4616 Lakewood Drive
    City Residence = Bingham
    Date of Death = 18 April 1976
```

If further information were given later on in the article (he had a wife and a son), these new facts would have to be added to the memory token for John Doe during Internalization. Internalization would also be responsible for recognizing that any references to 'the deceased,' 'Mr. Doe,' or 'Mrs. Doe's husband,' are in fact references to the same memory token.

Sometimes Internalization requires accessing knowledge structures. If after hearing 'John went to a restaurant,' the next sentence is 'He ordered a hamburger,' the initial parse will fill the Actor slot with HUMO where HUMO has the properties of being human and male. When this parse is internalized, the restaurant script which was activated by the first sentence is accessed to see if any of the conceptualizations predicted by the script applier match the current input. In this case a match would be found, and the match tells us that the actor of the current concept must be bound to the role of the patron in the restaurant script. If someone orders food in a restaurant, we expect that someone to be acting in the role of a restaurant patron. During the processing of the first sentence, the memory token for John was bound to the patron role. In this way the pronominal referent in the second sentence is determined to be John. HUMO from the initial parse is then replaced with a pointer to the memory token representing John.

Problems in Internalization affect question answering whenever references must be correctly recognized. Some question answering problems which are related to Internalization will be discussed in 3.1.3.3. But for the most part, QUALM is independent of Internalization processes. For a description of the Internalization program which is used by SAM and PAM, see [Cullingford 1977][1].

---

[1] Throughout this thesis conceptualizations in Conceptual Dependency presented as illustrations will be written in graphic form, with components as they would appear after the intial parse, before Internalization takes place. This is done at the expense of technical correctness for the sake of the reader, who I assume feels more at home with JOHN than G0001.

4. Conceptual Categorization

Conceptual Categorization is performed by the Question Analyzer. The Question Analyzer takes the internalized parse of a question and decomposes it into two descriptive components: a question concept and a conceptual question category. For example, the question:

Q5: Did John hit Mary?

has the conceptual question category 'Verification' and a question concept representing 'John hit Mary.' Question concepts are represented in Conceptual Dependency and are derived from the internalized parse of the question according to rules developed for each question category.

There are thirteen conceptual categories for questions. The Question Analyzer recognizes which category a question belongs to by running the question through a series of tests which function like a simple discrimination net [Feigenbaum 1963]. For example, Causal Antecedent questions are recognized by a test which checks for:

1) a causal chain construction
2) the causal link = LEADTO
3) all or part of the leading conceptualization is unknown

Conceptual question categories, the rules used to identify the conceptual category of a question, and the rules used to extract question concepts, are all described in Chapter Two.

5. Inferential Analysis

Very often the correct interpretation of a question involves understanding in terms of inferences. These inferences may rely on assumptions about the questioner's desires or goals, assumptions about what the questioner does and does not know, and assumptions about what is really being asked. All inferences of this sort are the result of higher memory processes which examine the question concept and its conceptual categorization in an effort to understand it beyond its literal meaning.

For example, if John is packing for a business trip and he asks his wife:

Q6: What haven't I packed?

His question is inferred to mean:

Q7: What haven't I packed that I should have packed?

Without this additional interpretation, Q6 will admit all sorts of ridiculous and useless answers.

In a similar way, many requests are recognized only after inference processes are invoked. For example:

Q8: Why don't you get Mary a drink?

can easily be meant as a request:

Q9: Would you get Mary a drink?

rather than as an inquiry for reasons behind not getting a drink. On the other hand,

Q10: Why don't you feel angry?

makes little sense as a request:

Q11: Would you (please) feel angry?

Inferential Analysis contains the inference processes which are essential to understanding what the questioner really wants to know. Without this understanding, many seemingly simple answers cannot be produced. Without Inferential Analysis of questions the following exchange would be impossible:

Q12: Does it snow in Portland?
A12: Maybe once every year or two.

If no inference-based interpretation is allowed, the information given in A12 would have to be painfully extracted:

Q13: Does it snow in Portland?
A13: Yes.
Q14: How often?
A14: How often what?
Q15: How often does it snow in Portland?
A15: Maybe once every year or two.

If a system had no capacity for the inferential interpretation of questions, question answering dialogs would progress slowly and deliberately like the one above. Rules for Inferential Analysis of questions are described in Chapter Three.

In Figure 1 the interpretive processes of parsing, conceptual categorization, and inferential analysis are outlined. A sample question (Do you have a dime?) is parsed into a Conceptual Dependency representation which is equivalent to the question 'Do you have a dime in your immediate possession?' The Question Analyzer (Chapter Two) then categorizes this question as a Verification question and extracts a question concept (You have a dime in your immediate possession). Inferential Analysis (Chapter Three) then recategorizes the question as a Request and reinterprets the question to mean 'Will you give me a dime?'

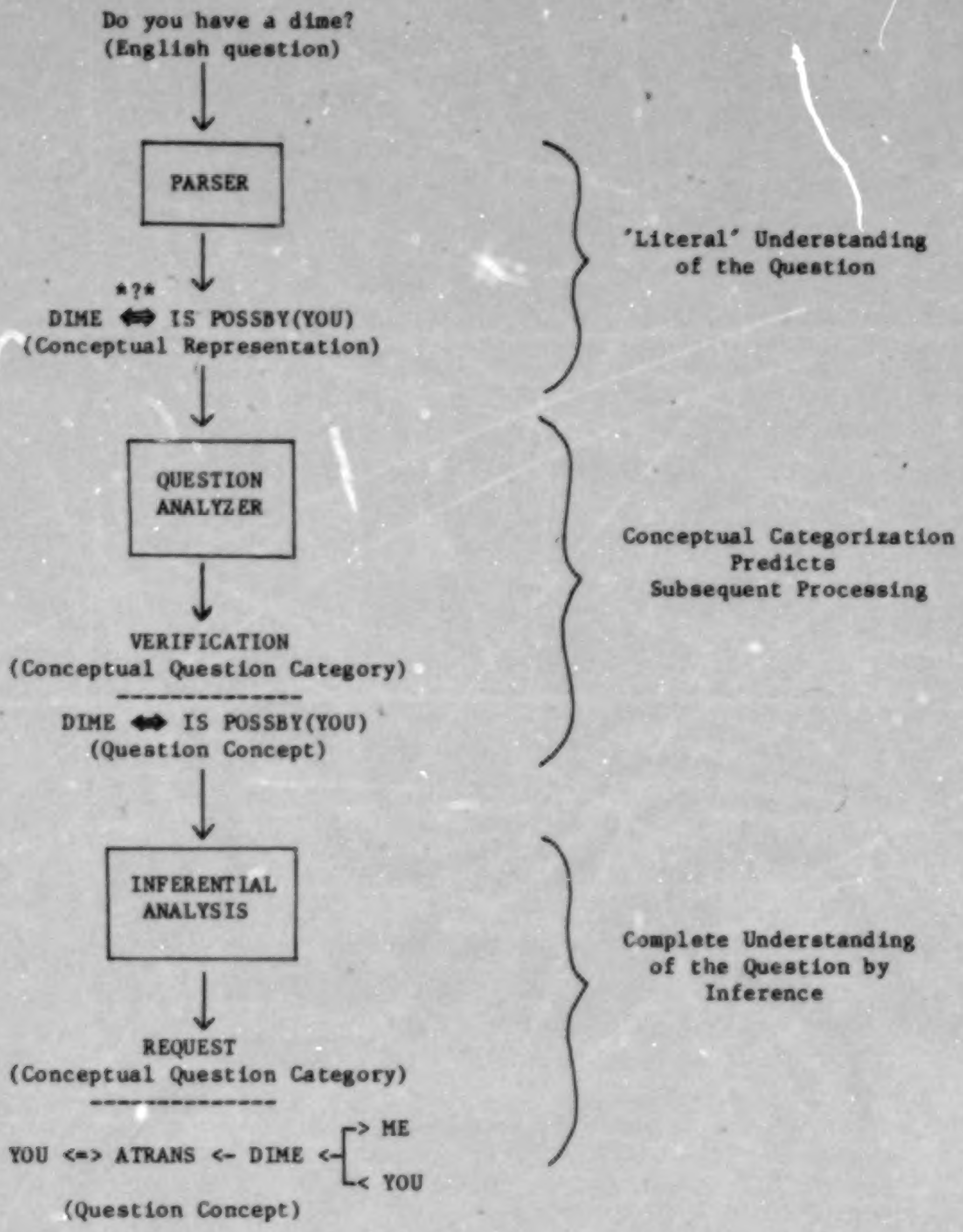

Do you have a dime?
(English question)
PARSER
?
DIME ⇔ IS POSSBY(YOU)
(Conceptual Representation)
QUESTION ANALYZER
VERIFICATION
(Conceptual Question Category)
DIME ⇔ IS POSSBY(YOU)
(Question Concept)
INFERENTIAL ANALYSIS
REQUEST
(Conceptual Question Category)
YOU <=> ATRANS <- DIME <-
-> ME
< YOU
(Question Concept)
'Literal' Understanding of the Question
Conceptual Categorization Predicts Subsequent Processing
Complete Understanding of the Question by Inference

Figure 1

Stages of Interpretation

CHAPTER 2

CONCEPTUAL CATEGORIES FOR QUESTIONS

This chapter describes the Question Analyzer within QUALM. The Question Analyzer examines the conceptual parse of a question and assigns a Conceptual Category to that question.

Q1: Why did John kill the dragon?

is a 'Causal Antecedent' question which could be answered with either:

A1a: Because the dragon took Mary.
A1b: To save Mary.

But if Q1 were worded a little differently, it would fall into a different Conceptual Category:

Q2: For what purpose did John kill the dragon?

Q2 is a 'Goal Orientation' question. As such, it does not elicit the same kinds of answers as a Causal Antecedent question. Q2 can be appropriately answered with A1b, but A1a is no longer a good answer:

Q2: For what purpose did John kill the dragon?
A1a: Because the dragon took Mary.

Conceptual Categorization recognizes essential conceptual differences in questions. These differences are ultimately reflected by the types of answers which are appropriate for a given question.

2.0 Introduction

In grammar school textbooks questions are often categorized lexically. There are who-questions, what-questions, where-questions, when-questions, why-questions, did-questions, and how-questions. Sometimes questions are categorized in terms of the grammatical part of speech which will provide an answer. So there are nominal questions and adverbial questions. A special case of answer-oriented categories are yes/no questions. These familiar categories do not constitiute a comprehensive system and are not motivated by anything greater than a desire to have a few general descriptive devices.

Lexical question categories seem to exist primarily for the purpose of textbook exercises. They are also used to describe parts of speech: e.g. an adverb is a word which answers a where, when, or how-question. But lexical question typing has one indisputable advantage: everyone can understand that a who-question is any question which begins with the word who. The concepts are sufficiently obvious to be universally understood. As long as these categories are intended to function descriptively for a general and non-technical audience, lexical typing is an effective and adequate system. In fact, throughout this thesis references to lexical categories of questions will be found in spite of the fact that these lexical categories are not recognized by QUALM and are not useful from a processing point of view.

In this chapter a system of thirteen conceptual categories for questions is presented. Subsequent references to question types will describe questions according to this conceptual category system whenever the technical process model is being discussed. When a non-technical reference is admissible, lexical categories are used for the sake of readability. If the reader has a passing familiarity with Conceptual Dependency decompositions [Schank 1972, 1973b, 1974a, 1975a] the relationships between lexical and conceptual question categories should be grasped with little difficulty.

When formulating a process model for question answering, a category system for questions must be descriptive in terms of that process model. If a given question falls into a particular category, that should tell us something about the processing which the question must undergo. To see how lexical typing fails in this respect, consider all the different kinds of how-questions there are:

Quantity

Q1: How long is this?
(What is the length of this?)
Q2: How often does this happen?
(What is the frequency of this occurrence?)

Q1 and Q2 are quantification questions. Each of these questions ask for a description of quantification requiring a measurement in units. How long is this? - 14 inches. How often does this happen? - Once every week or two. The units of quantification do not always have to be explicitly referenced. If Q2 is answered 'Seldom,' this answer must be interpreted against some norm of frequency.

Relative Description

Q3: How intelligent is John?
(What is the relative intelligence of John?)
Q4: How wet is your coat?
(What is the relative wetness of your coat?)

Q3 and Q4 are relative scale questions. These questions ask for a description along some scale (say -10 to 10) where there is a norm dependent on the nature of the property. While some people might chose to quantify intelligence in terms of IQ points, it is also acceptable to describe relative intelligence with terms like 'very bright,' or 'a little slow,' where these descriptions implicitly reference some assumed norm against which a comparison is being made.

Attitude

Q5: How do you like New York?
(What are your feelings about New York?)

Q5 asks for an attitudinal orientation. This question could be answered by specifying virtually any attitude imaginable: I hate it, I love it, I am totally unaffected by it, I try not to think about it, I wouldn't wish it on a dog, I can't wait to get out, I've found my niche, etc. etc. Appropriate answers to this question are more flexible than relative scale specification. 'I can't wait to get out,' tells us that the answerer expects to leave someday in addition to the fact that he has a negative attitude toward New York.

Emotional/Physical State

Q6: How do you like your eggs?
(What physical state do you prefer your eggs in?)
Q7: How is John?
(What is the emotional/physical state of John?)

Q6 and Q7 ask for state descriptions. Eggs can be over-easy, sunny-side-up, poached, fried, or scrambled. John could be just fine, on the critical list, morbidly depressed, euphoric, or he could have a slight cold. Very often answers to questions like this combine causal information: 'He is depressed about the stock market,' or 'He is excited about the new house.'

Enablement

Q8: How were you able to buy this without money?
(What enabled you to buy this?)
Q9: How did you get here so fast?
(What enabled you to arrive faster than I expected?)
Q10: How could you hear what he said?
(What enabled you to hear him?)

Q8, Q9, and Q10 ask about enabling conditions. Some state or act was a necessary enablement for the acts in question.

Instrumentality

Q11: How did you get here?
(By what means did you come here?)
Q12: How did you send word to him?
(By what means did you communicate to him?)

Q11 and Q12 ask about the instrumentality of the acts in question. In Q11 some transportational conveyance is sought and Q12 asks for a vehicle of communication.

Causal Antecedent

Q13: How did the glass break?
(What caused the glass to break?)

Q13 is asking for a causal antecedent. Something happened which caused the glass to break (it was dropped, hit, thrown, or crushed, etc.)

Instructions

Q14: How do I get to your house?
(Would you give me instructions to your house?)
Q15: How do you get service around here?
(What do you have to do to get service here?)

Q14 and Q15 ask for instructions. Answers to these questions often involve describing a chain of actions which must be executed in sequence.

The memory searches which will find answers to these questions vary considerably. Quantification questions require an examination of object properties in terms of a numerical measurement. Relative scale questions require a description relative to some norm on a comparative scale. Attitudinal orientation and state descriptions may be combinations of relative state scales and other information. Enablement questions require an examination of events which are causally related to the conceptual event in question. Instrumental questions ask for descriptive specification of events simultaneous in time with the act in question. Causal antecedent questions require knowledge of causal responsibility, and procedural specification questions require retrieval of instructional information.

A useful taxonomy of questions would predict the kinds of memory searches needed to answer any given question. It would also be useful if question categories determined which inference mechanisms have to be invoked for a complete interpretation of the question. A question category should predict the processes which are needed to understand and answer questions falling in that category.

In order to be useful as a predictive mechanism which effectively guides processing, categories must be assigned to questions before higher memory processes are summoned for further interpretation and memory searches. This means that a question category should describe its members in some manner which will allow us to assign the correct category to a question as soon as possible.

The earliest point at which categorization could take place is before the question is parsed, while the question exists only as a lexical entity. But we have seen that lexical categories are too misleading for process model predictions. So we cannot expect to assign question categories before the question has been parsed. This should not be viewed as a loss. The parsing processes for questions rely on the same predictive mechanisms which are applied to declarative statements [Riesbeck 1975, 1976]. The parser would not benefit from knowing that the question it was working on fell into one question category or another. But after the parse of a question is completed, the conceptual question is ready for higher interpretive memory processing. A question category should be recognized before the higher memory processes are invoked. Therefore question categories should be assigned immediately after a question is parsed.

In QUALM question types are assigned as soon as the parse is completed and a Conceptual Dependency representation has been produced for the question. Categorizing a question is the first task of the interpreter.

QUALM uses thirteen conceptual question categories:

1) Causal Antecedent
2) Goal Orientation
3) Enablement
4) Causal Consequent
5) Verification
6) Disjunctive
7) Instrumental/Procedural
8) Concept Completion
9) Expectational
10) Judgemental
11) Quantification
12) Feature Specification
13) Request

These question categories can be recognized by a simple examination of the conceptual question. Salient structural features of the conceptual question are examined by testing procedures which are hierarchically organized in the manner of a discrimination net [Feigenbaum 1963]. The Question Analyzer which performs this analysis will be described in 2.14. Once a question category has been assigned by the Question Analyzer, this categorization will be a central factor in subsequent interpretation and memory searches.

The Question Analyzer also establishes question concepts in addition to assigning conceptual categories to questions. A question concept is roughly what is left of a question when the interrogative aspect of the question is removed. For example, the question concept for 'Why did John go to New York?' is the conceptualization representing 'John went to New York.' Rules for extracting question concepts are different for each conceptual question category.

Categories for questions are only useful to the extent that they predict those processing strategies which will result in a correct interpretation and successful memory search. If two proposed categories require identical processing strategies, there is no rationale for distinguishing separate categories. Conversely, if a single proposed category requires many different processing strategies in order to effectively understand and answer all questions in that category, then it will be useful to split that category up into smaller ones which better predict the necessary processing. The conceptual categories proposed here can be thought of as processing categories which are predicted by features of conceptual representation.

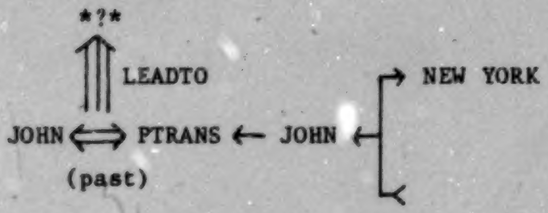
The justification for these thirteen categories will become more apparent when the processes predicted by them are described. Inferential Analysis, Content Specification, and Retrieval Heuristics are all processing modules within QUALM which rely on conceptual question categorization.

2.1 Causal Antecedent

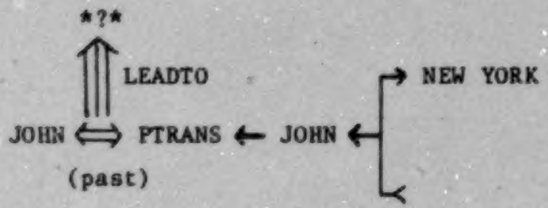
Causal Antecedent questions ask about states or events which have in some way caused the concept in question. Many different kinds of causal relationships are covered by Causal Antecedent questions (e.g. physical causality and motivating emotional states).

Examples:

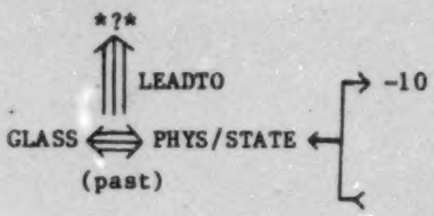
Q1: Why did John go to New York?



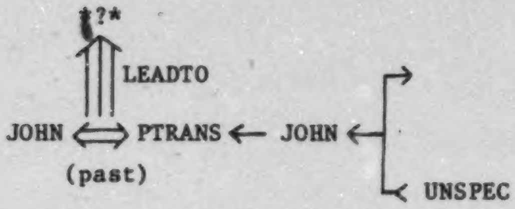
Q2: What caused John to go to New York?



Q3: How did the glass break?



Q4: What resulted in John's leaving?



Causal Antecedent questions are always represented as causal chain structures [Schank 1973a, 1974b, 1975b, Appendix 1], where the chain antecedent is unknown. Since the precise nature of the causal relationship is also unknown, the causal link between the unknown antecedent and the question concept is a LEADTO link.

Recognizing the Category:

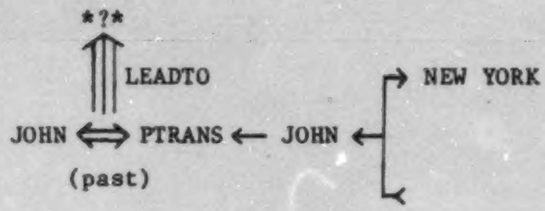
All Causal Antecedent questions are recognized by the following features:

(1) a casual chain of two conceptualizations
(2) causal link is LEADTO
(3) all or part of the first conceptualization is unknown

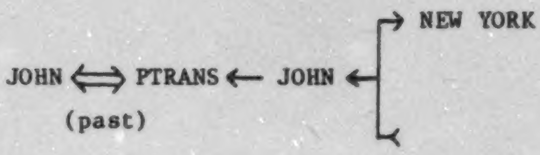
Finding the Question Concept:

The question concept of a Causal Antecedent question is extracted from the parsed question by deleting the first conceptualization in the causal chain.

For example:


?
LEADTO
JOHN
PTRANS
JOHN
NEW YORK
(past)

has the question concept:

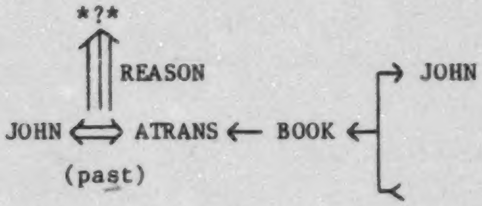

JOHN
PTRANS
JOHN
NEW YORK
(past)

2.2 Goal Orientation

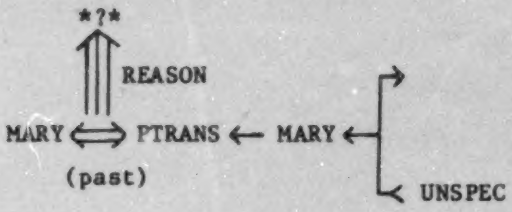
A Goal Orientation question is a special case of what is commonly called a why-question. Questions in this category may be paraphrased as why-questions, but Goal Orientation questions ask about the motives or goals behind an action. This makes them slightly more specific than Causal Antecedent questions. Since a mental goal is being sought as the explanation, the unknown causal antecedent relates to the question concept as a reason for the act in question; therefore the causal link between the unknown antecedent and the act in question is a REASON link.

Examples:

Q1: For what purpose did John take the book?


?
REASON
JOHN
ATRANS
BOOK
JOHN
(past)

Q2: Mary left for what reason?



Appropriate answers to Goal Orientation questions should describe the mental state of the actor in the question concept. This presupposes that the actor of the question concept is a human who acts of his own volition. Goal Orientation questions ask about the mental processes and desires underlying human behavior. It does not make sense to ask 'For what purpose did the book fall?' because the conceptual representation for a falling book has gravity PTRANSing the book, and gravity does not act out of volition but from laws of physics. 'For what purpose did John fall?' is similarly nonsensical since gravity is still the actor acting upon John and John presumably did not fall on purpose. In order for this question to make any sense at all we have to twist our usual understanding of what it means to fall; we assume that John feigned a fall or threw himself off balance on purpose. As soon as this element of intentionality is injected, it makes sense to ask what was John's reason for falling.

Some Causal Antecedent questions can be answered in terms of either a Causal Antecedent or a Goal Orientation:

Q3: Why did Mary drop the book?
A3a: Because John bumped her.
(causal antecedent)
A3b: To get John's attention.
(goal orientation)

A3a describes an act which RESULTed in Mary dropping the book while A3b describes a REASON Mary had for dropping the book. When the question is understood as a Causal Antecedent question (as would be the case for Q3) either answer can be returned. But if the question were worded as a Goal Orientation question (For what purpose did Mary drop the book?) the answer would have to describe a reason for the act in question. It does not make sense to answer:

Q4: For what purpose did Mary drop the book?
A3a: Because John bumped her.

In section 5.2 we will see how QUALM looks first for a Goal-Oriented answer and then settles for a more general Causal-Antecedent answer if no Goal-Oriented answer is found. The issues of how to answer why-questions when there are many reasonable responses is dealt with at length in Chapter Eight.

Recognizing the Category:

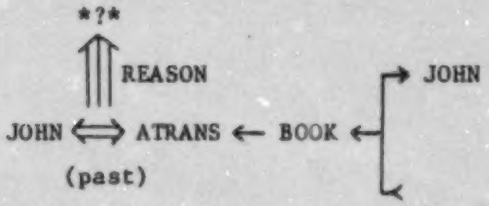
All Goal Orientation questions are recognized by the following features:

(1) a casual chain of two conceptualizations
(2) causal link is REASON
(3) first conceptualization is unknown

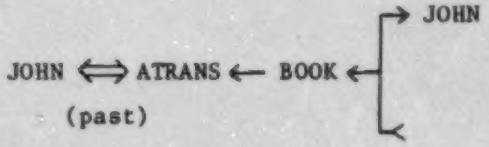
Finding the Question Concept:

The question concept of a Goal Orientation question is extracted from the parsed question by taking the second conceptualization in the causal chain.

For example:



has the question concept:

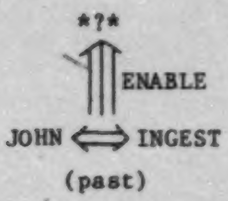


2.3 Enablement

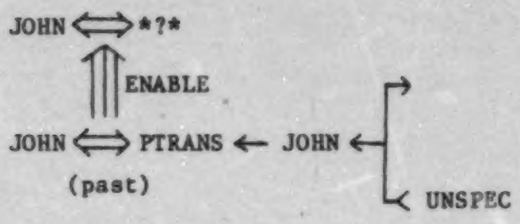
Enablement questions are similar to Goal Orientation questions insofar as they specify a causal relationship between an unknown conceptualization and the question concept. The causal relationship is an ENABLE and the concept in question is enabled by the unknown act or state.

Examples:

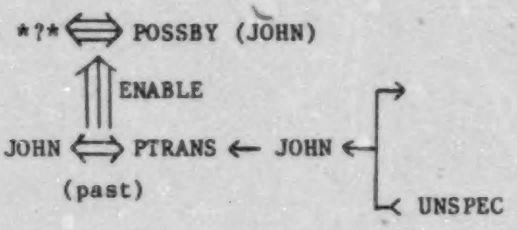
Q1: How was John able to eat?


?
ENABLE
JOHN
INGEST
(past)

Q2: What did John need to do in order to leave?


JOHN
?
ENABLE
JOHN
PTRANS
JOHN
UNSPEC
(past)

Q3: What did John need to have in order to leave?


?
POSSBY (JOHN)
ENABLE
JOHN
PTRANS
JOHN
UNSPEC
(past)

Enablement questions may specify some information about the unknown concept. Q2 describes the unknown enablement as an action while Q3 describes it to be a state of immediate possession. In general, an Enablement question consists of a question concept enabled by another concept which is either completely unknown or which has an unknown component.

The precise nature of the enabling relationship is left unspecified. Enablements can occur in a variety of ways. Eugene Charniak [Charniak 1975b] has explored different kinds of enabling causation. Two of his categories are physical enablement (John needed a car to go to New York) and social enablement (John needed money to eat at the restaurant). World knowledge about cars and money and restaurants is needed to determine when enablements are physical and when they are social. In SAM and PAM there is only one causal link describing Enablement: the ENABLE link.

If more than one Enablement link were used within these systems, there would have to be some process somewhere which relied on such distinctions. If no such process exists, there is no justification for having different causal links. Thus far, SAM and PAM have had no need for different Enablement links.

The inference processes which operate on Enabling relationships appear to be very knowledge specific. For example, if you go to a restaurant without money, you will be waited on and served without any difficulty until the check comes. Then the consequences of having no money can include having to wash dishes or being arrested. If you go to a restaurant without shoes on, you may very well be turned away at the door. But you won't be arrested or made to wash dishes. Knowing that the enablements violated in these two cases are both social enablements does not in itself help us know what the specific consequences of violating those conditions are. Only very specific knowledge about restaurants will allow us to predict the consequences of not having money or not wearing shoes.

Recognizing the Category:

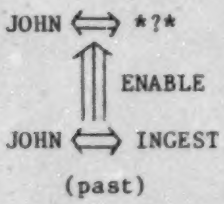
All Enablement questions are recognized by the following features:

(1) a casual chain of two conceptualizations
(2) causal link is ENABLE
(3) all or part of the first conceptualization is unknown

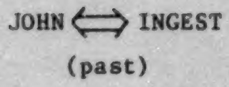
Finding the Question Concept:

The question concept of an Enablement question is extracted from the parsed question by taking the second conceptualization in the causal chain.

For example:



has the question concept:

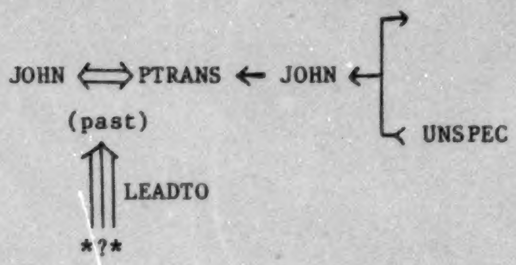

JOHN
INGEST
(past)

2.4 Causal Consequent

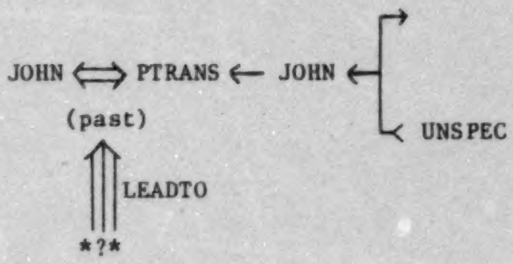
Causal Consequent questions are causal structures in which the question concept causes an unknown concept or causal chain in some way. The general causal link for such questions is the LEADTO link.

Examples:

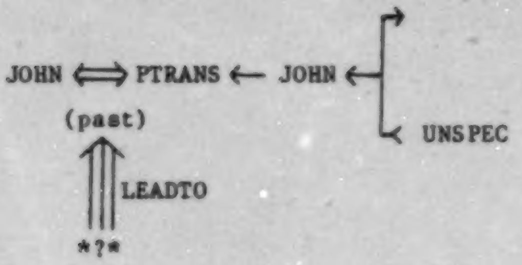
Q1: What happened when John left?


JOHN
PTRANS
JOHN
UNSPEC
(past)
LEADTO
?

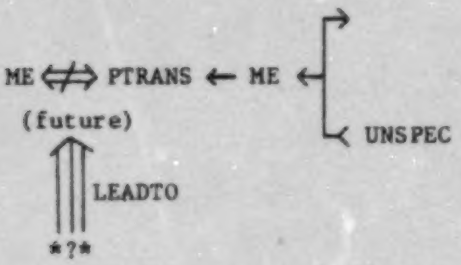
Q2: What resulted from John leaving?


JOHN
PTRANS
JOHN
UNSPEC
(past)
LEADTO
?

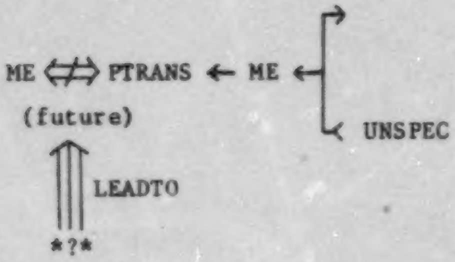
Q3: What happened after John left?


JOHN
PTRANS
JOHN
(past)
UNSPEC
LEADTO
?

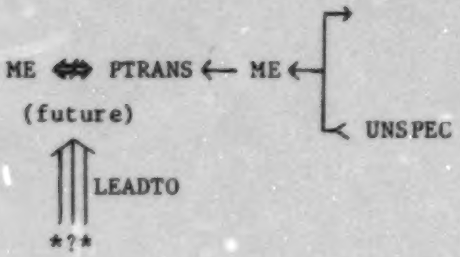
Q4: What if I don't leave?


ME
PTRANS
ME
(future)
UNSPEC
LEADTO
?

Q5: What happens if I don't leave?

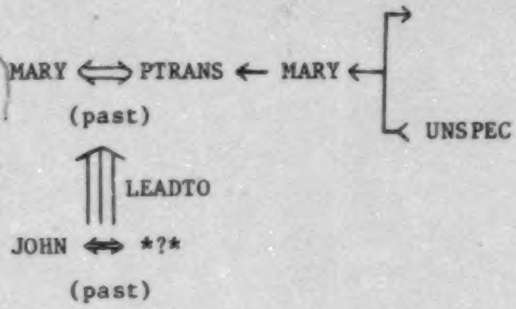

ME
PTRANS
ME
(future)
UNSPEC
LEADTO
?

Q6: If I don't leave, then what?


ME
PTRANS
ME
(future)
UNSPEC
LEADTO
?

Some Causal Consequent questions have partially known consequents:

Q7: What did John do after Mary left?


MARY
PTRANS
MARY
(past)
UNSPEC
LEADTO
JOHN
?
(past)

Recognizing the Category:

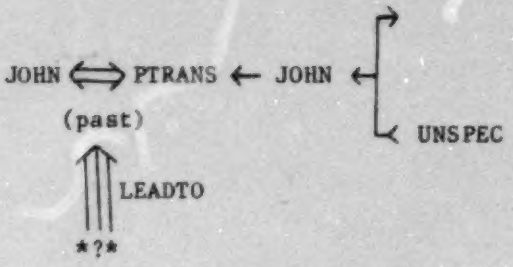
All Causal Consequent questions are recognized by the following features:

(1) a casual chain of two conceptualizations
(2) causal link is LEADTO
(3) all or part of the second conceptualization is unknown

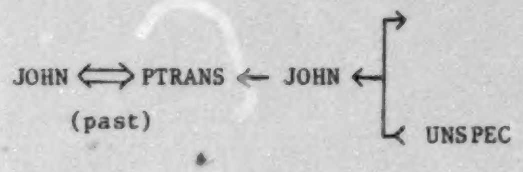
Finding the Question Concept:

The question concept of a Causal Consequent question is extracted from the parsed question by taking the first conceptualization in the causal chain.

For example:


JOHN ⟺ PTRANS ← JOHN
(past)
UNSPEC
LEADTO
?

has the question concept:

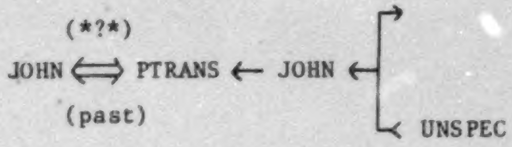

JOHN ⟺ PTRANS ← JOHN
(past)
UNSPEC

2.5 Verification

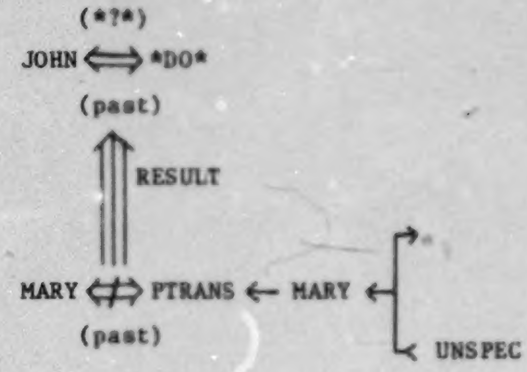
Verification questions ask about the truth of an event. These questions correspond roughly to those questions which can be answered yes or no. They are represented as single concepts or as causal chain constructions with a MODE value = *?*.

Examples:

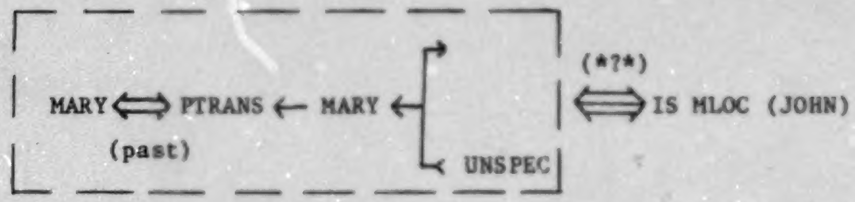
Q1: Did John leave?


(*?*)
JOHN ⟺ PTRANS ← JOHN
(past)
UNSPEC

Q2: Did John do anything to keep Mary from leaving?


(*?*)
JOHN
DO
(past)
RESULT
MARY
PTRANS
MARY
(past)
UNSPEC

Q3: Does John think that Mary left?


MARY
PTRANS
MARY
(past)
UNSPEC
(*?*)
IS MLOC (JOHN)

Recognizing the Category:

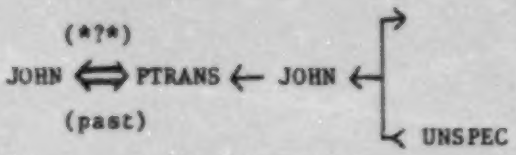
All Verification questions are recognized by one of the following features:

(1) A single conceptualization with MODE = *?*, or
(2) A causal chain construction containing a
conceptualization with MODE value = *?*.

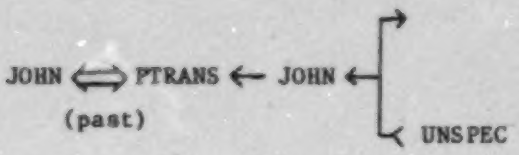
Finding the Question Concept:

The question concept of a Verification question is extracted from the parsed question by removing the MODE value *?* from the conceptualization.

For example:


(*?*)
JOHN
PTRANS
JOHN
(past)
UNSPEC

has the question concept:

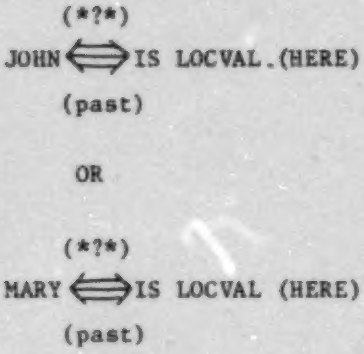

JOHN
PTRANS
JOHN
(past)
UNSPEC

2.6 Disjunctive

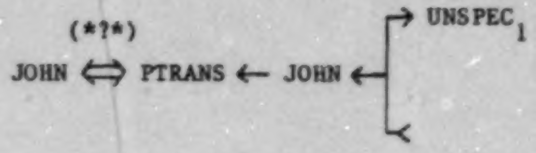
Disjunctive questions are like Verification questions but with multiple question concepts instead of one.

Examples:

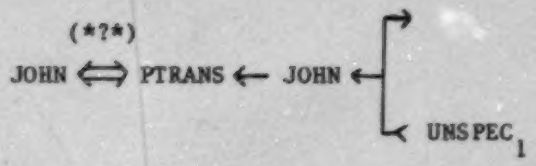
Q1: Was John or Mary here?


(*?*)
JOHN
IS LOCVAL.(HERE)
(past)
OR
(*?*)
MARY
IS LOCVAL (HERE)
(past)

Q2: Is John coming or going?


(*?*)
JOHN
PTRANS
JOHN
UNSPEC$_1$

OR


(*?*)
JOHN
PTRANS
JOHN
UNSPEC$_1$

Recognizing the Category:

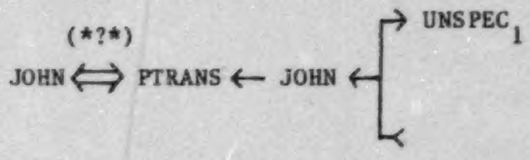
All Disjunctive questions are recognized by the following features:

1. A top-level OR relation
2. Concepts under the OR relation have MODE = *?*

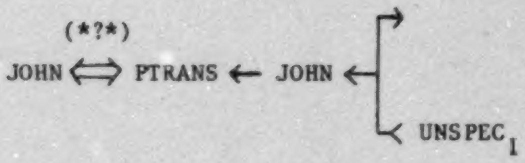
Finding the Question Concept:

The question concept of a Disjunctive question is extracted from the parsed question by listing the conceptualizations without their MODE value = *?*.

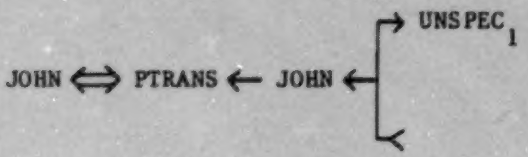
For example:

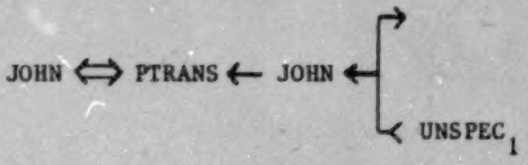

(*?*)
JOHN
PTRANS
JOHN
UNSPEC$_1$

OR


(*?*)
JOHN
PTRANS
JOHN
UNSPEC$_1$

has the multiple question concept:


UNSPEC
1
JOHN
PTRANS
JOHN


JOHN
PTRANS
JOHN
UNSPEC
1

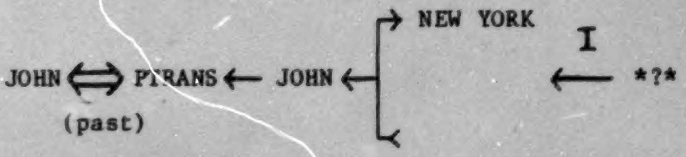
While Disjunctive questions are much like Verification questions but with multiple question concepts, the processing for these questions is distinct from the processing for Verification questions. It is rarely the case that a Disjunctive question can be appropriately answered with a yes or no.

2.7 Instrumental/Procedural

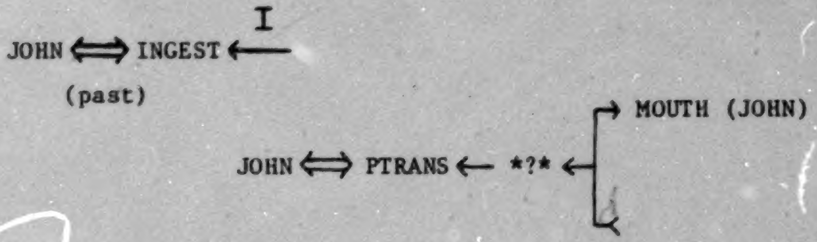
Instrumental/Procedural questions are represented by concepts which have a totally or partially unknown instrumentality:

Examples:

Q1: How did John go to New York?


NEW YORK
I
JOHN
PTRANS
JOHN
?
(past)

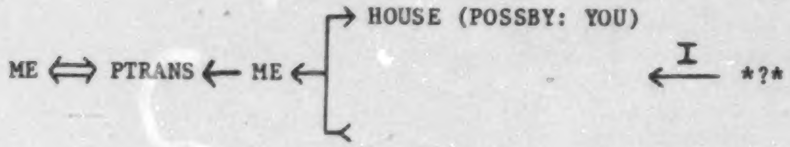
Q2: What did John use to eat with?


I
JOHN
INGEST
(past)
MOUTH (JOHN)
JOHN
PTRANS
?

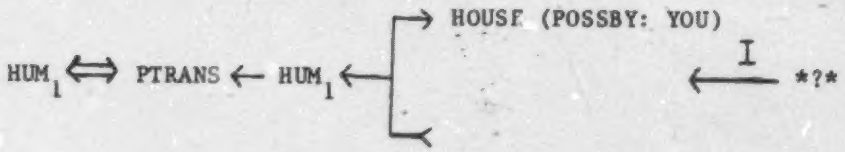
Sometimes the instrumentality for an act entails a long sequence of acts rather than a single event. In this case, the unknown instrument is more appropriately described as a procedure. Procedural questions are represented in the same way as instrumental questions. The difference lies in what kind of answers are expected; a procedural question is looking for directions, and an instrumental question is looking for a short answer.

Examples:

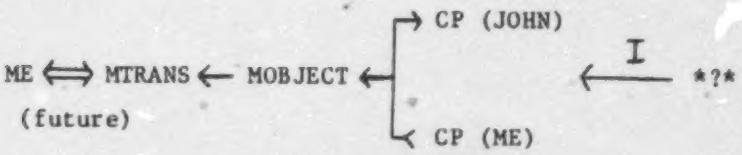
Q3: How do I get to your house?



Q4: What is the best way to your house?



Q5: How will I get word to John?



Whether or not a question is looking for an instrumental or a procedural answer is a decision which must be made by interpretive memory processes according to the specific context, the questioner's assumed knowledge state, and other inference processes. The initial parse of a question is not responsible for deciding what kind of answer is most appropriate for a given question. There are contexts in which the question 'How did John get to your house?' will be best satisfied with an answer like 'By car.' But there are also contexts (e.g. if the obvious route has been altered by a system of complicated detours) in which the question asks for a more detailed description. Inferential Analysis and Content Specification are responsible for deciding how much information is being asked for by an Instrumental/Procedural question.

There are many how-questions which may look like they should be Instrumental/Procedural questions but are not:

Q6: How did you see John?
(Did you make an appointment? threaten him?)
Q7: How did you find your lost book?
(Did you offer a reward? Look for it yourself?)
Q8: How can we eat out tonight?
(Can we cash a check somewhere? Use a credit card?)

Each of Q6-8 ask about actions or conditions which must precede the act in question. An Instrumental/Procedural question asks about an act which was simultaneous with the main act of the question. If a question asks about an act which precedes the main act of the question in time, the question is either a Causal Antecedent or an Enablement question.

Q6: How did you see John?

asks what steps had to be taken before you were able to see John. It does not ask for acts which were simultaneous with the act of seeing John (acts like talking to him or watching him).

Q7: How did John find his lost book?

Finding a lost object is conceptually represented as a change in mental state. Finding a lost book means that the location of the book was unknown (not accessible to the actor's Conscious Processor) for some period of time, but then the location became known. 'How did John find his lost book?' asks for an act which resulted in this mental state change. John may have found the book by asking Bill if he knew where it was or by looking through his desk drawers. In any case, the act precedes the state change as a causal antecedent.

Q8: How can we eat out tonight?

If this question were taken as an Instrumental/Procedural question it could be answered 'We'll use forks and knives.' The question is much more likely to be asking about the enabling conditions for eating out. Reasonable answers are along the lines of 'We can borrow Bill's car,' or 'I have an American Express Card.' Each of these answers specifies an act or state which will enable the act of eating out.

Knowledge about when questions like Q6-8 should be understood as Causal Antecedent or Enablement questions instead of Instrumental/Procedural questions is not something which the parser can always be expected to have. The correct interpretation of questions like these often occurs at a higher level of interpretive analysis. Rules for reinterpreting Instrumental/Procedural questions as Causal Antecedent or Enablement questions are incorporated in the Inferential Analysis described in Chapter Three.

Recognizing the Category:

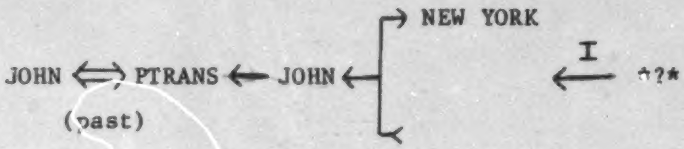
All Instrumental/Procedural questions are recognized by the following feature:

1. The question involves a conceptualization which has a partially or totally unknown instrument.

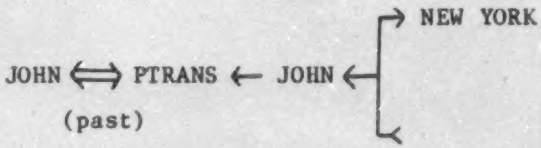
Finding the Question Concept:

The question concept of an Instrumental/Procedural question is extracted from the parsed question by removing the Instrument slot.

For example:



has the question concept:

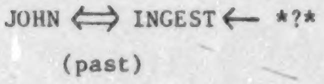


2.8 Concept Completion

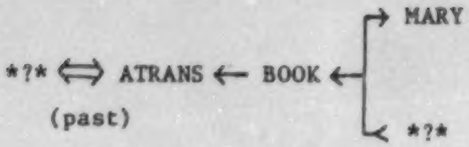
Concept Completion questions include a lot of who, what, where, and when-questions. These questions are very much like fill-in-the-blank questions insofar as they specify a particular event with one missing component and ask for the completion of that component.

Examples:

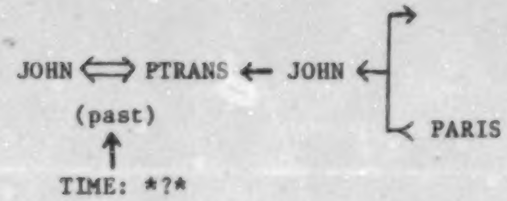
Q1: What did John eat?



Q2: Who gave Mary the book?



Q3: When did John leave Paris?



Recognizing the Category:

All Concept Completion questions are recognized by the following feature[1]:

1. An unknown conceptual component somewhere in the question conceptualization.

Finding the Question Concept:

The question concept for a Concept Completion question is identical to the parsed question conceptualization.

---

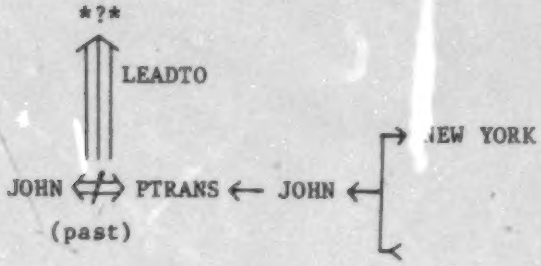
[1] This description holds unless the conceptual question satisfies the specifications of another conceptual category in which case the other category has precedence over Concept Completion. For example, if the unknown component is the Instrument slot filler, the question is Instrumental/Procedural.

2.9 Expectational

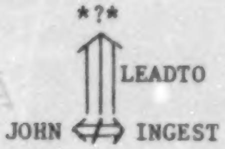
Expectational questions ask about the causal antecedent of an act which presumably did not occur. This presupposition is what sets Expectational questions apart from Causal Antecedent questions. Expectational questions are usually phrased as why-not questions.

Examples:

Q1: Why didn't John go to New York?



Q2: Why isn't John eating?



Recognizing the Category:

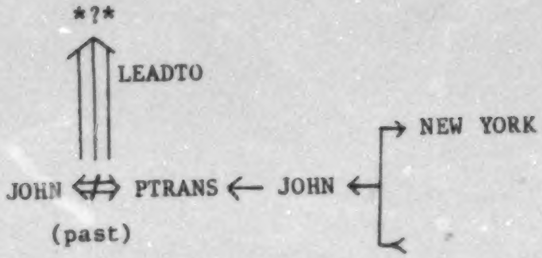
All Expectational questions are recognized by the following features:

1. A causal chain of two concepts
2. The first concept is unknown
3. The causal link is LEADTO
4. The second concept has a MODE value = *NEG*

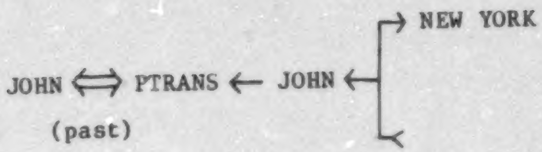
Finding the Question Concept

The question concept of an Expectational question is extracted from the parsed question by taking the second concept from the causal chain and deleting its negative MODE value.

For example:



has the question concept:



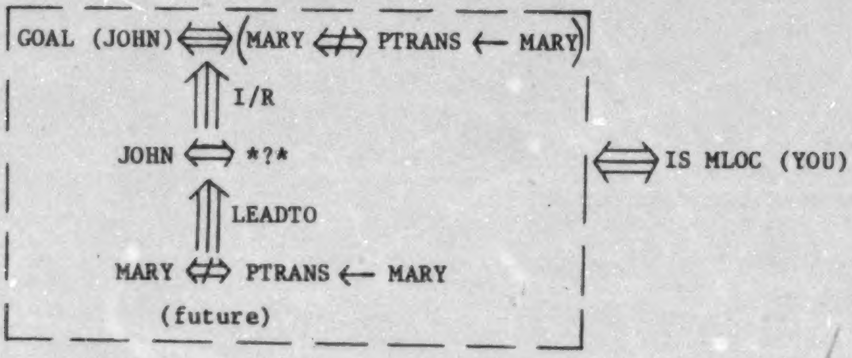
2.10 Judgemental

Judgemental questions are those which solicit a judgment on the part of the listener. All Judgemental questions can be appropriately prefaced by 'In your opinion...' Of course all questions ask for an opinion of the questioner, so such a distinction could be viewed as nothing more than a matter of degree. But without getting into difficult philosophical arguments, Judgemental questions are roughly those questions which require a projection of events as opposed to the strict recall of facts. 'Where is St. Louis?' is not a question which requires a Judgemental answer. This question asks for a hard fact which a person either knows or doesn't know. 'Where do you expect the President will spend Christmas?' is not asking for a hard fact unless the answerer is known to be a close friend of the President who knows about all of his personal plans.

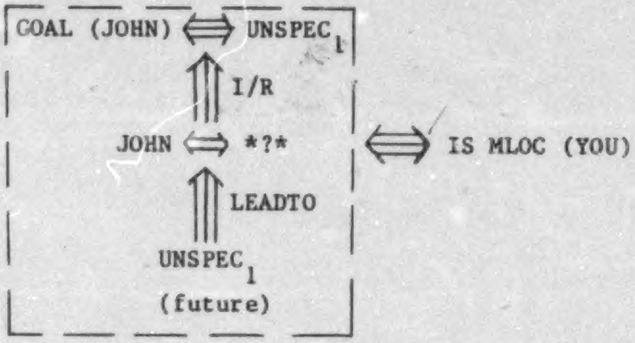
Judgemental questions are recognized by their explicit reference to the mind of the person being addressed.

Examples:

Q1: What should John do to keep Mary from leaving?


GOAL (JOHN) ⟺ (MARY ⟺ PTRANS ← MARY)
I/R
JOHN ⟺ *?*
LEADTO
MARY ⟺ PTRANS ← MARY
(future)
⟺ IS MLOC (YOU)

Q2: What should John do now?


GOAL (JOHN) ⟺ UNSPEC1
I/R
JOHN ⟺ *?*
LEADTO
UNSPEC1
(future)
⟺ IS MLOC (YOU)

Recognizing the Category:

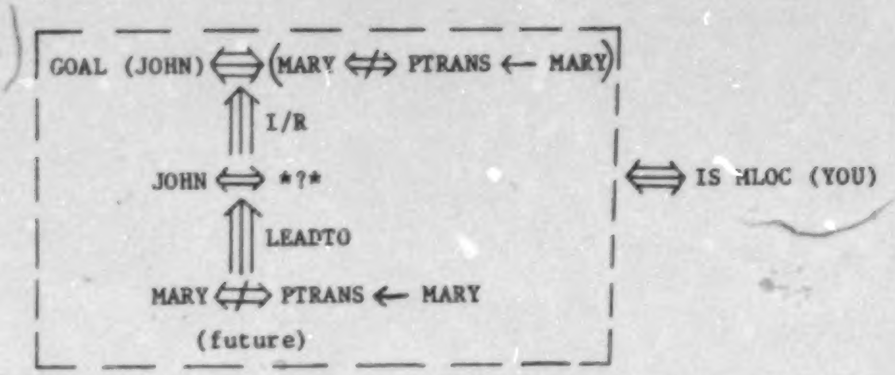
All Judgemental questions are recognized by the following features:

1. A top level MLOC state
2. The top level actor is the answerer.
3. The MOBJECT contains a conceptualization which is partially unknown.

Finding the Question Concept:

The question concept for Judgemental questions is extracted from the conceptual question by finding the goal state inside the MOBJECT. If no goal is specified, a default goal must be derived from the context in which the question is asked.

For example:


GOAL (JOHN) ⟺ (MARY ⇎ PTRANS ← MARY)
I/R
JOHN ⟺ *?*
LEADTO
MARY ⇎ PTRANS ← MARY
(future)
⟺ IS MLOC (YOU)

has the question concept:

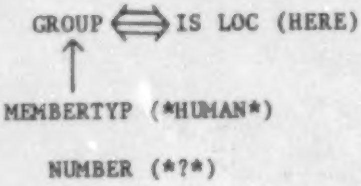
GOAL (JOHN) ⟺ (MARY ⇎ PTRANS ← MARY)

2.11 Quantification

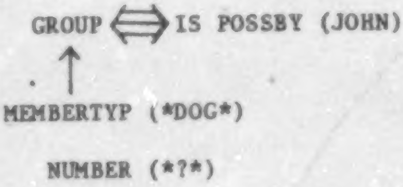
Quantification questions are those which ask for an amount. The amount may be countable as in how-many questions or it may be a continuous quantity. All Quantification questions can be phrased as how-questions.

Examples:

Q1: How many people are here?


GROUP ⟺ IS LOC (HERE)
MEMBERTYP (*HUMAN*)
NUMBER (*?*)

Q2: How many dogs does John have?


GROUP ⟺ IS POSSBY (JOHN)
MEMBERTYP (*DOG*)
NUMBER (*?*)

Some Quantification questions refer to values on physical or mental state scales. In this case, the amount in question is a relative value on a finite scale.

Examples:

Q3: How ill was John?

```
JOHN ⟺ IS PHYSSTATE (*?*)
  (past)
```

Q4: How badly do you want the book?

```
BOOK ⟺ POSSBY (YOU)
      ⇑ CANCAUSE
 YOU ⟺ TOWARD JOY (*?*)
```

Q5: How does John feel?

```
JOHN ⟺ IS MENTALSTATE (*?*)
```

Recognizing the Category:

All Quantification questions are recognized by the following features:

1. A causal chain or single conceptualization involving a state scale
2. An unknown state scale value

Finding the Question Concept:

The question concept for a Quantification question is identical to the parsed question conceptualization.

2.12 Feature Specification

Feature Specification questions ask about some property of a given person or thing. Feature Specification questions are similar to Concept Completion questions insofar as they are both fill-in-the-blank type questions. The significant difference between the two question types is that Concept Completion questions ask about missing conceptual components in actions while Feature Specification questions ask about static properties of objects.

Examples:

Q1: What color are John's eyes?

$$\begin{tikzcd} \text{EYES} \arrow[r, Leftrightarrow] & \text{IS COLOR (*?*)} \\ \text{POSSBY (JOHN)} \arrow[u] & \end{tikzcd}$$

Q2: What breed of dog is Rover?

$$\begin{tikzcd} \text{ROVER} \arrow[r, Leftrightarrow] & \text{IS BREED (*?*)} \end{tikzcd}$$

Feature Specification questions ask about properties which cannot be expressed as a relative value on a scale. For example, colors are conceptualized in terms of names. While it is possible to represent colors on a wave length spectrum, people do not naturally think about color in this manner. But it would be misleading to say that Feature Specification questions look for names while Quantification questions look for relative numerical assignments. There are some Feature Specification questions which are answered in terms of numerical quantities:

Q3: How much does that rug cost?

$$\begin{tikzcd} \text{RUG} \arrow[r, Leftrightarrow] & \text{IS COST (*?*)} \end{tikzcd}$$

Q4: How old is John?

$$\begin{tikzcd} \text{JOHN} \arrow[r, Leftrightarrow] & \text{IS AGE (*?*)} \end{tikzcd}$$

These questions can be anwered in terms of a number but they are Feature Specification questions because any numerical answer given must refer to a specific unit. The unit need not be explicitly stated in the answer, but it is nevertheless there implicitly by inference. If John's age is stated to be 56 we would infer that he is 56 years old. If the rug is said to cost a hundred, we assume it costs $100. It may cost 100 sheep but we would infer our own standard monetary unit unless told otherwise.

Feature Specification questions which ask about non-numeric properties can refer to implicit properties in the same way that the units of numeric properties are often inferred. This happens when questions are phrased 'What kind of ...' or 'What sort of ...'

Q5: What kind of dog is Rover?
Q6: What kind of doctor is John?
Q7: What sort of college is this?
Q8: What sort of bicycle does John have?

By inference most people would interpret Q5 to be asking about the breed of dog while Q6 asks for a branch of medical practice. Q7 would normally be understood to be asking about educational orientation while Q8 could be asking about the make or general type. These questions are open to flexible interpretations which should be sensitive to context. In different contexts, Q5 could be referring to breed (beagle), variety (hound), or lifestlye (housedog). But in any given context, some inference must be made about what property is being sought in order to answer the question.

Recognizing the Category:

All Feature Specification questions are recognized by the following features:

1. There is an unknown property value

Finding the Question Concept:

The question concept for a Feature Specification question is identical to the parsed question conceptualization.

2.13 Request

Requests constitute a special question category which is distinct from all the other question categories presented here. All of the other question categories discussed in this chapter describe inquiry questions. An inquiry is asked by a questioner who is seeking some specific information. All inquiries are appropriately answered via an MTRANS of some sort. But a Request is asked when the questioner wants a specific act to be performed.

Q1: Would you pass the salt?
Q2: Can you get me my coat?
Q3: Will you take out the garbage?

It is not adequate to characterize Requests and inquiries in terms of whether or not a verbal response is appropriate in response. 'What time is it?' is an inquiry which can be answered by pointing to a clock on the wall. And many requests are denied verbally:

Would you pass the salt?
No.

Even requests which are performed are often accompanied by a verbal response:

Would you pass the salt?
Sure, here.

Requests are different from inquiry question types in terms of when they are recognized by the interpreter. Inquiry question types are initially recognized by the Question Analyzer. But Requests are never recognized by the Question Analyzer. All Request questions are assigned some inquiry question type by the Question Analyzer. It is then up to the Inferencial Analysis to reassign the question category as a Request. This distinction in recognition derives from the fact that all requests can be literally interpreted as inquiries.

Q1: Would you pass the salt?
Q2: Can you get me my coat?
Q3: Will you take out the garbage?

Each of these questions could be (mis)understood as a Verification question answerable by a Yes or No. The Question Analyzer will always understand a Request literally. So it will recognize Q1-3 as Verification questions instead of Requests. The Inferential Conversion Rules are responsible for finding the less literal interpretations of questions which capture what the questioner 'obviously' meant. It is on this higher level of question interpretation that Requests are detected by means of interpretive inference mechanisms.

2.14 The Question Analyzer

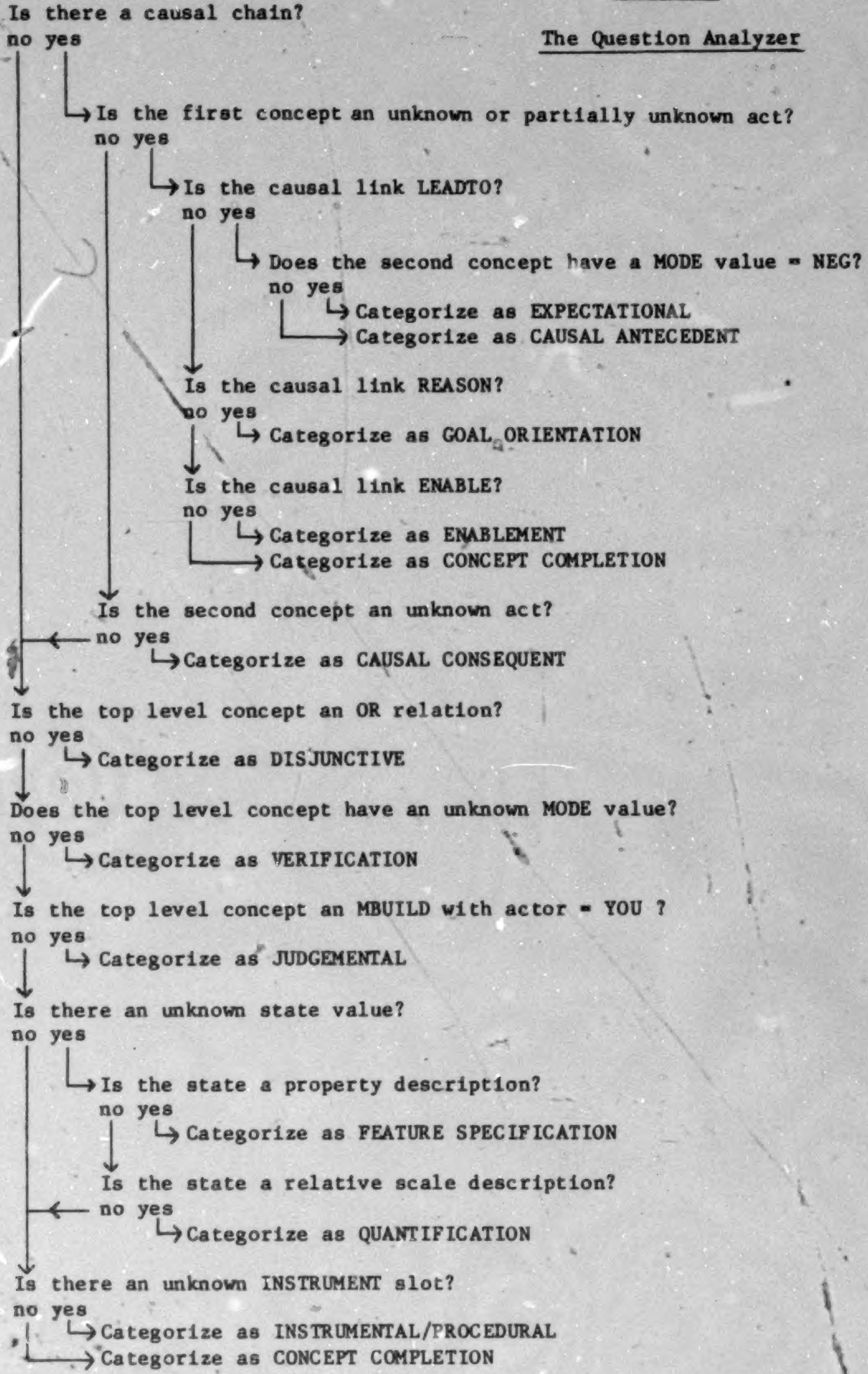
The Question Analyzer is designed to function like a discrimination net which applies various tests to a conceptualized question in order to determine its Conceptual Category. The tests within the net are hierarchically organized in order to minimize the test processing. For example, at the head of the net is a test which determines whether or not a question is represented as a causal chain structure. If it is, then tests for Causal Antecedent, Goal Orientation, Enablement, Causal Consequent, and Expectational questions are all organized under this one branch of the net.

The most difficult category to recognize are the Concept Completion questions. These questions are characterized by an unknown conceptual component which may occur anywhere in the conceptualization, at any level. Rather than conduct a search to positively identify such questions, the net is organized so that any question which has passed through a terminal branch without positive identification will be categorized as Concept Completion.

Figure 2 outlines the organization of the discriminating tests used by the Question Analyzer.

Figure 2

The Question Analyzer


Is there a causal chain?
no yes
Is the first concept an unknown or partially unknown act?
no yes
Is the causal link LEADTO?
no yes
Does the second concept have a MODE value = NEG?
no yes
Categorize as EXPECTATIONAL
Categorize as CAUSAL ANTECEDENT
Is the causal link REASON?
no yes
Categorize as GOAL ORIENTATION
Is the causal link ENABLE?
no yes
Categorize as ENABLEMENT
Categorize as CONCEPT COMPLETION
Is the second concept an unknown act?
no yes
Categorize as CAUSAL CONSEQUENT
Is the top level concept an OR relation?
no yes
Categorize as DISJUNCTIVE
Does the top level concept have an unknown MODE value?
no yes
Categorize as VERIFICATION
Is the top level concept an MBUILD with actor = YOU ?
no yes
Categorize as JUDGEMENTAL
Is there an unknown state value?
no yes
Is the state a property description?
no yes
Categorize as FEATURE SPECIFICATION
Is the state a relative scale description?
no yes
Categorize as QUANTIFICATION
Is there an unknown INSTRUMENT slot?
no yes
Categorize as INSTRUMENTAL/PROCEDURAL
Categorize as CONCEPT COMPLETION

CHAPTER 3

RECATEGORIZING QUESTIONS BY INFERENTIAL ANALYSIS

This chapter describes inference mechanisms within QUALM which complete the interpretation of a question. When the Question Analyzer assigns a Conceptual Category to a question, it does so on the basis of structural features in the Conceptual Dependency representation of that question. Inferential Analysis examines the content of a question to see if this initial categorization of the question is correct.

In the context of talking to a friend on the street, the two questions:

Q1: Do you have a match?
Q2: Do you have a wooden match?

are both recognized by the Question Analyzer to be 'Verfication' questions. That is, they are both understood as inquiries deserving of yes or no answers. But while Q2 is best understood as an inquiry, Q1 should ultimately be understood as a request for a light. Q2 should remain an inquiry because the specification of a wooden match as opposed to any other kind of match suggests that the questioner is interested in something other than merely getting a light. Q1, on the other hand, is a standard way of requesting a light.

By the time the Inferential Analyzer examines Q1 and Q2, they have both been tagged as Verification questions by the Question Analyzer. It is the job of Inferential Analysis to change the category for Q1 to a 'Request,' and further specify that Q1 is a request for a light (in the sense of a flame).

3.0 Introduction

Many questions are not correctly understood if taken literally. Some questions require knowledge about conversational conventions in order to be interpreted correctly, and others can only be understood within their situational context. Many questions can best be answered by taking into consideration what the questioner knows and doesn't know. All of these factors contribute to the final interpretation of a question. The interpretive analysis of a question must be able to perceive the question within its overall environment; interpretive mechanisms must be sensitive to a wide range of information which is 'external' to the question. These mechanisms must effectively examine a question, consider contextual factors, appeal to the conventions of dialog, and conclude that 'at this time, in this setting, this question must mean (blitch).' Interpetations which do this require understanding by inference.

Inference mechanisms are needed to achieve complete conceptual understanding of a question. When a question is not interpreted correctly on the level of Inferential Analysis, the answer produced will be inappropriate. Faulty inferential analysis of a question results in an answer which may have been right in another setting, but not the current one.

There are three inference modules within the interpretor which are designed to recognize what a question is really asking. These three modules contain:

(1) Contextual Inference Conversion Rules
(2) Context-Independent Inference Conversion Rules
(3) Knowledge State Assessment Conversion Rules

Each module contains rules of inference which enable the system to alter its understanding of a question. These rules are applied whenever the conceptual category and question concept meet specific criteria. When a conversion rule is applied to a question, it alters the conceptual category assigned to that question and usually alters the question concept as well. Each question is tested by all of the conversion rules and reinterpretation occurs whenever a rule is applicable. This process of successive reinterpretation continues until all the conversion rules have run their tests. The resulting question concept and conceptual categorization represent the final interpretation of the question.

3.1 Contextual Inferences

Contextual inference rules exploit the conversational context in which a question occurs. Conversational context refers to the situational setting in which a conversation takes place. There are three types of inference mechanisms which rely on conversational context: conversational scripts, generalized inference mechanisms, and conversational continuity.

Conversational Scripts

Some conversational settings are very stereotyped and conversations within these settings can be understood by invoking an appropriate script. An inference rule which is specific to one particular script is a conversational script inference. For example, the conversations which a stockbroker has with his clients all have predictable elements which center around the transactions which a stockbroker can perform. If one is not familiar with the transactions which normally occur in this particular business setting, a dialog between a stockbroker and his client cannot be fully understood. Conversational scripts are knowledge structures which organize knowledge-based inferences for specific situations where people interact conversationally.

Generalized Inference Rules

Not all knowledge-based inferences can be organized under specific situational scripts in the way conversational scripts organize inference mechanisms. If John has been helping Mary pack for a trip, and he asks:

> Q1: What didn't I pack?

Mary will understand this question to mean:

> Q2: What didn't I pack that I should have packed?

There is no conversational script for conversations about packing since dialogs about packing are not highly stereotyped. But there is a situational script about packing for trips which must be used to understand the question. In QUALM this knowledge is accessed and utilized by a generalized inference rule. This rule is general in the sense that it would be equally applicable in a context where John is mailing out Christmas cards and he asks Mary:

> Q3: Who didn't I send a card to?

Here again there is no conversational script which goes with sending Christmas cards, but there is a script for sending Christmas cards which must be accessed to understand the question. The same general inference rule which operated in the context of packing for a business trip will work in the context of sending Christmas cards. This inference conversion rule will reinterpret Q3 to mean:

> Q4: Who didn't I send a card to who I wanted to send a card to?

Without this inference, the question would admit all sorts of ridiculous answers:

> Q3: Who didn't I send a card to?
> A3: Abraham Lincoln, Moby Dick, and everyone in Nova Scotia.

The general rule which handles questions like Q1 and Q3 is described in 3.1.2.2 A general rule of this sort which utilizes script-related knowledge but which can be applied to a variety of scripts is called a generalized inference conversion rule.

Conversational Continuity

A final application of conversational context occurs when a question relies on the continuity of conversation. Many questions do not contain complete conceptualizations but make sense because they are understood in terms of previous conversation. Conversational continuity conversion rules are responsible for understanding in these situations.

A: Did John give the book to Mary?
B: No.
A: Why not?

In this case it is clear that the question 'Why not?' is asking 'Why didn't John give the book to Mary?'

A: Did John give the book to Mary?
B: I don't know.
A: Why not?

Now 'Why not?' means 'Why don't you know if John gave the book to Mary?' General rules of conversational continuity allow us to fill in missing information like this.

3.1.1 Conversational Scripts

Many conversations which occur between strangers in the context of an everyday business or service transaction are highly predictable in terms of content. A clerk in a store spends the majority of his time answering questions like:

How much is this?
Do you have this in another color (size, style)?
Do you expect to get any more in?
Do you have any ...?
Where are the ...?

Of course a customer can ask a store clerk anything (Haven't we met before?) but questions which are appropriate to the role a store clerk assumes are both finite in nature and highly predictable. As for the store clerk, he is expected to do nothing but answer questions and initially offer assistance (Can I help you?).

The memory processes which predictively anticipate stereotypic exchanges within a common situation are encoded in a conversational script. The term script, as it has been presented within the SAM system, has been used primarily to refer to a predictive mechanism which has knowledge about stereotypic sequences of actions. Technically speaking, this type of script is more correctly described as a situational script [Schank & Abelson 1977] since it describes what events are liable to occur in a given situation. A conversational script is a predictive mechanism which has knowledge about stereotypic conversations.

Many situational scripts have conversational scripts embedded in them. For example, the restaurant script should contain a conversational script for dialogs between the patron and waiter/waitress. The restaurant script implemented in the SAM system has a minimal version of a complete conversational script. There is an MTRANS conceptualization in which the patron tells the waiter what he wants, and there is an branch of events off of the main path in which the waiter may MTRANS to the patron that he can't have what was ordered. This last MTRANS may be followed by an MTRANS from the patron to the waiter specifying another order. This is as much as SAM knows about patron/waiter dialogs in the ordering scene of the restaurant script.

If SAM had a complete conversational restaurant script it would be able to understand stories where the patron asks the waiter to describe a specific dish, or to make a recommendation. Social conventions often determine what is and is not included in a conversational script. It's acceptable for a waiter to praise your choice when you place an order ('Oh yes, that's a very good dish'), but it's unlikely that he will tell you when you're making a mistake ('I think that's our worst dish - are you sure you won't reconsider?'). If a waiter tells you that you've made a poor choice, you feel that he is acting outside of his role as a waiter.

Conversational scripts are used in question answering whenever a situation gives rise to stereotyped exchanges. If after consulting a menu in a restaurant, the waiter comes up with pad in hand and says:

Are you ready to order now?
What'll it be?
Are you ready here?
Have you decided yet?
Do you want some more time?
Yes?

All of these questions will be readily understood to be asking for an order. Anyone who has eaten out in a foreign country without any knowledge of the native language will attest to the fact that a waiter who appears with pad in hand at the right point in the script can say just about anything and still get an order. Scriptal situations affect the interpretation of questions just as they affect all script-governed conversation.

In some stereotypic settings conversational scripts dictate how things are said as well as what things get said. Conversational scripts can affect the stylistic rules of conversation. Style here refers to attitudinal styles such as formality (business meetings), casualness (parties), or sobriety (funerals). Stylistic features of questions are important because an inappropriate style can constitute a statement in itself. For example, excessively polite inquiries at a party are often used to discourage further conversation.

In addition to stylistic rules, conversational scripts often specify something more on the order of subcultural dialects. A good example of this occurs at auctions where the auctioneer utilizes a highly stylized mode of conversation while interacting with the floor. The rules of conversation in this context extend well beyond verbal conversation into visual signals for bidding, but there are still rules for interpreting verbal questions which must be known in order to understand the interaction. These rules allow us to understand a variety of questions.

Do I hear $50? -> Will someone bid $50?
Who will give me $50? -> Who will bid $50?
Is there $50 out there? -> Will anyone bid $50?

In this case the auctioneer could say virtually anything which had a reference to a number in it and the audience would understand it to be a reference to either the bid currently standing or the bid the auctioneer would like to get:

> I'm at $40, I want $50.
> I've got $40 in the hand, is there $50 in the bush?
> I found $40, I'm looking for $50.

Each of these statements can be easily understood because the conversational script for an auction predicts that an auctioneer will tell the audience two things during bidding: the standing bid and the bid he wants.

Inference mechanisms based on conversational scripts must be script-specific. The interpretive rules for an auctioneer during bidding are very simple:

> (1) If two numbers are mentioned, the lower one is the standing bid and the higher one is the bid sought.
>
> (2) If one number is mentioned, it is the bid sought.

These rules are applied whenever the meaning of a statement does not explicitly reference the standing bid or the bid sought. These interpretive rules are useless outside of an auction. If John is thinking about buying a $50 chair and he asks Mary 'Do we have $50?' Mary will not interpret the question to mean 'Will you bid $50?' The interpretive mechanisms described above should be accessed only when an auctioneer is talking to his audience during bidding.

The notion of conversational scripts and their role in interpretive processing must be included in any question answering model which claims to be a general model. As far as computer implementation of QUALM is concerned, there is no need to implement conversational scripts in the question interpreter unless we expect the computer to be carrying on conversations in various settings (restaurants, stores, bars, etc.). As long as the computer is confined to the context of answering questions about stories, it has no need to access conversational scripts during question interpretation. In the context of story undertanding, conversational scripts are needed during understanding (if a story contains any script-related dialog) but not during question answering. John could ask Mary 'Your place or mine?' but nobody's liable to put that question to the computer.

Since SAM and PAM have not dealt with stories in which there is any direct dialog, no conversational scripts have been implemented in these systems. There has, however been some work done which is closely related to the notion of conversational scripts and their application to conversational programs. The application of script derived predictions has been explored (quite promisingly) in the domain of airline reservations with the development of a program called GUS [Bobrow et al. 1976]. While this work has been conducted independently and without reference to the general notion of scriptal conversation, it is clear that GUS relies on script-specific knowledge. GUS assumes the role of an airline reservation agent in the conversational context of a reservationist talking to a client. GUS's script-related predictions guide it through dialogs where it must determine where the customer wants to go, when, and for how long, etc. We will discuss GUS and its relationship to QUALM more fully in Chapter Eleven.

3.1.2 Generalized Inferences

Many questions require inferencing in order to fill in missing information. These inferences are often made on the basis of scriptal knowledge about the world. When an inference mechanism needs to access knowledge from a script, there are basically two ways that the mechanism may be designed. Some rules which generate inferences by drawing on script-based knowledge can be stated very generally so that one rule will apply over a large set of scripts. Other rules are script-specific and can be used in the context of one script for which they were designed. This notion of general applicability vs. script-specific applicability distinguishes generalized inference mechanisms from conversational script inference mechanisms. We will describe three generalized inference rules here. These rules tend to be very powerful because they can be used in a variety of contexts.

3.1.2.1 Single Word Questions

In any script where there is a host or servant type of role, a question by the person in that role which merely specifies an object or list of objects means 'Do you want some (grobs)?' where (grobs) are something normally offered to the consumer in that particular script.

> Coffee, tea, or milk? (on an airplane)
> Nuts? (from someone fixing you a hot fudge sundae)

The rule about offering objects which applies to these examples can be stated even more generally:

> Rule #1:
>
> If a role in a script specifies a highly predictable act associated with an object, a question by a person in that role which specifies that object means 'Do you want me to (blitch)?' where (blitching) is the act predicted by the script.

So a butler can ask a guest who has just entered the house 'Your coat, sir?' and this will be understood to mean 'Do you want me to take your coat?' If the guest is about to walk out the door and the butler walks up to him carrying his coat, 'Your coat, sir?' means 'Don't forget your coat,' or 'May I help you with (help you put on) your coat?' Similarly, a waitress can go up to a man who has just sat down at a table, ask 'A menu?' and mean 'Do you want a menu?'

But suppose the man who is sitting at the table says to the waitress 'A menu?' Now the question means 'Would you bring me a menu?' So if a question in a script situation specifies an object associated with the script, the question should be interpreted as either an inquiry concerning the script act or a request that the script act be performed. Whether the question is an inquiry or a request depends on whether or not the questioner is the script assigned performer of the act.

So this leads us to the final most general formulation of the rule:

> Rule #2:
>
> A question which consists of a single noun is either a request or an inquiry about the script assigned act associated with that object. It is an inquiry about whether or not the act is desired when the questioner is playing the script role which performs the act in question. It is a request for the act to be performed when the questioner is anyone else.

When a script contains multiple events involving an object, each of which is performed by the same actor (as is the case with a butler and a coat) the appropriate act is determined by the current predictions of the script applier [Cullingford 1976, 1977]. If John has just entered a wealthy home, the script applier predicts that the butler may take his coat, but it does not predict that the butler will give John his coat back until later.

3.1.2.2 Universal Set Inference

It does not always make sense to word a Concept Completion question negatively. 'Who wasn't elected president in 1969?' is a fairly strange question in any context. But there are reasonable Concept Completion questions which ask about negated concepts. For example, 'What didn't I pack?' and 'Who haven't I invited?' are Concept Completion questions with negative MODE values since they ask about things which haven't been done. The interpretation of such questions always requires the addition of a constraining feature. Without such a constraint, these questions could be answered:

Q: What haven't I packed? (upon closing the suitcase)
A: This pile of fuzz and the World Trade Center.

Q: Who isn't here? (upon entering a college seminar)
A: Lassie, Kahlil Gibran, and Rosemary Woods.

Q: What haven't I added? (before baking cake)
A: A pound of dog hair and an oil filter.

It is obvious that all of these questions implicitly place a constraint on potential answers. While the nonsense answers may be absolutely correct, they are nonsense because they violated the implicit constraints: What haven't I packed (that I should have packed)? Who isn't here (who should be here)? What haven't I added (that belongs in the cake)? There is one general rule which can be used to specify these constraints on the basis of scriptal knowledge about the world. The rule is invoked whenever a Concept Completion question is encountered which has negative MODE value. The appropriate interpretive constraint is derived from the set of potential objects specified by the active script for the act or state in question. The script for packing a suitcase specifies that you pack only those objects which you want to have with you for reasons of necessity or convenience or peace of mind. The script for a college seminar specifies that the professor and registered students attend. The script for baking a cake specifies a set of ingredients.

When a negative Concept Completion question is asked, acceptable answers specify things from the script-defined set of potential considerations. If you don't know what the instantiated script-defined limitations are, you can't answer the question. Someone who doesn't know who belongs in the seminar can't hope to answer 'Who's not here?' In this way scripts delimit the set of acceptable and reasonable answers to negative specification questions. The more specific a script is, the easier it is to find an answer since the set of potential answers will be well defined. 'What haven't I packed?' is difficult to answer because it entails knowing what the person packing needs or would like to take with him. A person answering this question needs to know the person who is packing and where he is going in order to come up with good answers. It is much easier to answer 'What haven't I added?' in the context of baking a cake since the ingredients for baking a cake are specified by the script as the ingredients listed in the recipe.

3.1.2.3 Implicit Requests

Many social interactions between people are very common and have come to be fairly standardized. Requests which occur within a standard interaction are often phrased in a manner which is not altogether straightforward:

> Do you have a match?
> Do you know what time it is?
> Can you tell me what city this is?

In normal conversation, questions of this sort are never taken literally. That is, no one would ever answer one of these with a simple yes or no unless they were deliberately trying to be difficult. Each of these questions is in fact a request for a specific act or for the communication of information. 'Do you have a match?' is a request for a match. 'Do you know what time it is?' is an inquiry about the time, and 'Can you tell me what city this is?' is an inquiry about the name of a city. The literal interpretation of these questions does not convey their underlying meanings. But how are these underlying meanings recognized? The sentence structure or conceptual syntax of these questions is not sufficient to indicate when a question is an inquiry and when it is a request:

Do you have a Porsche?
(is not asking for the car)

Can you spell?
(is not asking for a demonstration)

We will now describe inference conversion rules which transform Verification questions into Requests whenever a Request interpretation is appropriate. Each rule has two parts: Criteria and Target Interpretation. The Criteria describe conditions under which the rule is appropriate. A rule will not be applied if its Criteria are not met. If all of the Criteria for a rule are satisfied, the question is reinterpreted by replacing the question category and question concept as specified by the Target Interpretation.

Some of the Criteria specifications require facts from general world knowledge concerning the content of the question. Therefore permanent memory may need to be accessed in order to verify the Criteria. These verification processes may be easy to execute or fairly involved. For example, the ATRANS Request Conversion asks whether or not the object of the conceptualization is of little value. This can be easily confirmed or denied by examining a memory token. But the Performance Request Conversion asks whether or not it is reasonable to ask that the act described by the question be performed. This is considerably more difficult to establish. The mechanisms which verify these Criteria will not be described. At this stage it is not clear how the more involved verification processes should be designed. If all of the Criteria for a rule apply, the question is reinterpreted according to the instructions under the Target Interpretation.

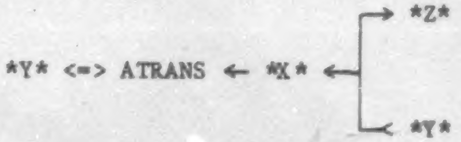
****

Rule 1: ATRANS Request Conversion

****

Criteria: (1) Conceptual Categorization = Verification
(2) The Question Concept is of the form:
X ⟺ IS POSSBY (*Y*) TIME (*PRESENT*)
where *Y* is the person being addressed
(3) *X* is of little value

Target Interpretation: Conceptual Categorization <- Request
Question Concept <-



where *Z* is the person asking the question

*****

Examples:

Rule 1 Applies:

Do you have a pencil? ---> Would you give me a pencil?

Do you have a quarter? ---> Would you give me a quarter?

Do you have a cigarette? ---> Would you give me a cigarette?

Rule 1 Does Not Apply:

Do you have a coat? (a coat is too valuable to give away)

Did you have a dime? (wrong tense)

Do you have a telephone? (a telephone is too valuable)

****

Rule 2: MTRANS Request Conversion

****

Criteria: (1) Conceptual Categorization = Verification
(2) Question Concept is of the form:
(*X*) $\Longleftrightarrow$ IS MLOC (*Y*) TIME (*PRESENT*)
where *Y* is the person being addressed
and *X* is a conceptualization involving an
unknown conceptual component
(3) the unknown component in *X* can be MTRANSed
quickly and easily

Target Interpretation: Conceptual Categorization <- Specification
Question Concept <- *X*

*****

Examples:

Rule 2 Applies:

Do you know what time it is? ---> What time is it?

Do you remember Al's address? ---> What is Al's address?

Do you recognize this song? ---> What song is this?

Rule 2 Does Not Apply:

Do you know John? (this sense of know is not represented by an MLOC construction with an unknown MOBJ)

Do you know how to spell? (no unknown component; spell what?)

Does Mary know where John is? (does not address respondent)

Did you know how to integrate trigonometric functions? (cannot be MTRANSed easily or quickly)

****

Rule 3: Performance Request Conversion

****

Criteria: (1) Conceptual Categorization = Verification
(2) Question Concept is of the form:
Y <=> DO_1 MODE (*CAN*)
TIME (*PRESENT*)
where *Y* is the person being addressed
and *DO_1* is some conceptual action

(3) Performance of the conceptual act DO_1 is a reasonable request

Target Interpretation: Conceptual Categorization <- Request
Question Concept <- *Y* <=> DO_1

*****

Examples:

Rule 3 Applies:

Can you give me a ride? ---> Would you give me a ride?

Could you light the fire? ---> Would you light the fire?

Rule 3 Does Not Apply:

Could you have lit the fire? (wrong tense)

Are you giving me a book? (MODE value ≠ *CAN*)

Can John tell Mary? (does not address the respondent)

The third criterion is stated vaguely because the necessary conditions which render a given act 'reasonable' are often dependent on the conversational context and the goals of the speaker. For example, in a small hospital where two doctors are conferring over an emergency, the question 'Can you perform open heart surgery?' may very well be a request to perform an operation. But the same question addressed to a doctor at a cocktail party will not be understood as a request.

****

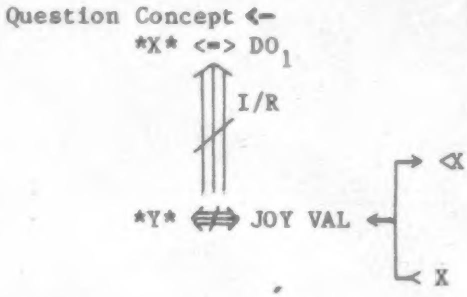
Rule 4: Permission Request Conversion

****

Criteria: (1) Conceptual Categorization = Verification
(2) Question Concept is of the form:
X <=> DO_1 MODE (*[illegible]*)
TIME (*PRESENT* or *FUTURE*)
where *Y* is the person asking the question
and DO_1 is some conceptual action

(3) the person addressed has power over *X*
with respect to the occurrence of DO_1

Target Interpretation: Conceptual Categorization <-
Verification



where *Y* is the person being addressed

*****

Examples:

Rule 4 Applies:

Can I go to the movies tonight? --->
Is it all right if I go to the movies tonight?

Can I take the car today? --->
Is it all right if I take the car today?

The third criterion is stated strongly and should be relaxed to cover social settings or relationships where both parties are equal. In such a situation it is reasonable to ask for permission in the weak sense of making sure there aren't any objections. For example, a man could ask his wife 'Can I take the car today?' without needing her permission in the sense of her allowing it. He could be asking just to make sure she hadn't planned on using it herself. This sense of a request is very different from one made in a relationship of unbalanced power (e.g. a teenager asking a parent for the car).

****

Rule 5: Function Request Conversion

****

Criteria: (1) Conceptual Categorization = Verification
(2) Question Concept is of the form:
X $\Longleftrightarrow$ POSSBY (*Y*) TIME (*PRESENT*)
where *Y* is the person being addressed
(3) The common function associated with *X*
can be easily executed

Target Interpretation: Conceptual Categorization $\Leftarrow$ Request
Question Concept $\Leftarrow$ *Y* <=> DO$_1$

where DO$_1$ is the instrumental script
commonly associated with *X*

*****

Examples:

Rule 5 Applies:

Do you have a light? ---> Would you offer me a light?

Rule 5 Does Not Apply:

Do you have a piano? (playing the piano is not trivial)

A transformation very similar to this one but slightly more complex goes into effect when additional information is given about a desired state change. For example:

Do you have air conditioning? It's too hot in here.
should be reinterpreted as
Would you turn on the air conditioning?

Do you have a light? It's dark in here.
should be reinterpreted as
Would you turn on a light?

The application of these conversion rules must be carefully ordered so that some rules will have precedence others. Consider the request for a cigarette:

Q: Do you have a cigarette?

If the Function Request Conversion were applied before the ATRANS Request Conversion, this question would be interpreted to ask 'Would you smoke a cigarette?' But if the ATRANS Request Conversion is applied first, the intended meaning ('Would you give me a cigarette?') is obtained.

****

The inference mechanisms described in this section are context dependent in the sense that contextual information may suppress them. For example, suppose John has been out drinking and he calls Mary up around 3 am to tell her he'll be home late. If in this context Mary asks John 'Do you know what time it is?' she is probably not asking for the time. The preceding rules will work most of the time, but not always. Higher level predictive processes can interfere with them and suppress them.

3.1.2.4 Implicit Causality

How-questions are subject to wide variations in conceptual representation. At the beginning of Chapter Two we saw eight distinct conceptual senses for questions beginning with the word how. Many of these senses are recognized by the parser, but others are left for higher memory processes to discern. In Chapter One (sections 1.1.3 - 1.1.4) we saw how in some contexts the question, 'How will we eat tonight?' should be understood as an Enablement question while in other contexts it is asking about Instrumentality. The parser does not attempt to determine which causality is more appropriate for a question like this. The parser will always represent 'How will we eat tonight?' as an Instrumental/Procedural question.

There are two inference mechanisms within Inferential Analysis which determine when an Instrumental/Procedural question should be reinterpreted as a Causal Antecedent or Enablement question: the Causal Antecedent Conversion and the Enablement Conversion. The Causal Antecedent Conversion operates independent of the context in which a question is asked and so it will not be presented until section 3.2. But the Enablement Conversion is sensitive to context and so will be presented now.

****

Rule 6: Enablement Conversion

****

Criteria: (1) Conceptual Categorization = Instrumental/Procedural

(2) Question Concept is of the form:
X <=> DO$_1$

where DO$_1$ describes a script instantiation

(3) One or more of the Enabling conditions for DO$_1$ are not satisfied

Target Interpretation: Conceptual Category ← Enablement
Question Concept ← *X* <=> DO$_1$

*****

Examples:

Rule 6 Applies:

How did you write this book? (you can't write) --->
How were you able to write this book?

How will we eat out tonight? (without any money) --->
How will we be able to eat out tonight?

How did John eat at Leone's? (since John was barefoot) --->
How was John able to eat at Leone's?

Rule 6 Does Not Apply:

How did John go to New York? (no violated enablements)

How are you contacting John? ('contacting' does not describe a script)

3.1.3 Conversational Continuity

Questions which occur in dialog are often incomplete. But people rarely have difficulty understanding what the intended question is in these cases. There are rules of conversational continuity which enable people to conceptually complete a partial question. When information is left out in dialog, conversational continuity is usually maintained by filling in information from the last conceptualization communicated. In this section a basic rule for question interpretation is presented which relies on conversational continuity. Two additional rules are given as variations on the basic rule which deal with pronominalization and incomplete questions.

3.1.3.1 The Basic Rule

Jane: Did John go to class yesterday?
Bill: No.
Jane: Why not?

Jane: Did John go to class yesterday?
Bill: I don't know.
Jane: Why not?

In the first dialog 'Why not?' refers to the fact that John didn't go to class yesterday. In the second dialog 'Why not?' refers to the fact that Bill didn't know if John went to class yesterday. There is one simple rule which explains how to complete an incomplete question concept which has been asked within dialog.

```
***************************************************
*                                                 *
*   Continuity Completion Rule:                   *
*                                                 *
*      Complete a partial question in terms of the *
*                                                 *
*      last concept communicated to the questioner *
*                                                 *
***************************************************
```

In the first dialog, the last thing the questioner heard before asking 'Why not?' was 'No.' Since this was an answer in response to a question, it communicated some conceptual information to the questioner. In this case, it conveyed the information that John did not go to class yesterday. Given the incomplete question 'Why not?' the rule tells us to combine 'Why not?' with the concept of John not going to class yesterday to get the complete question 'Why didn't John go to class yesterday?' In the second dialog, the last thing the questioner heard was 'I don't know.' This was also an answer in response to an earlier question, an answer conveying 'I don't know if John went to class yesterday.' So by using the rule, we combine 'Why not?' with 'I don't know if John went to class yesterday,' to get 'Why don't you know if John went to class yesterday?'

This rule works for a variety of incomplete questions. In addition to Causal Antecedent and Expectational questions, it works for many others as well:

A: Does it rain in Portland?
B: Yes. (it rains in Portland)
A: How much? (How much does it rain in Portland?)

A: Are you leaving for New York?
B: Yes. (I am leaving for New York)
A: Today? (Are you leaving for New York today?)

A: Is John still in charge?
B: No. (John is not still in charge)
A: Says who? (Who says John is not still in charge?)

3.1.3.2 The Last MLOC Update

The Last MLOC Update (LMU) is a temporary storage buffer which enables a question answerer to remember the last concept he communicated to the questioner. Each time a new exchange occurs in dialog, the LMU is updated with the last concept communicated and the previous contents of the LMU are lost. This simple device is sufficient for all of the conversational continuity inference rules.

Example:

Q: Does it rain in Portland?
A: Yes. (LMU = It rains in Portland)
Q: How much?

Q: Did Mary get the book from Susan?
A: No. (LMU = Mary did not get the book from Susan)
Q: Who did?

Q: Did John leave for New York?
A: Yes. (LMU = John left for New York)
Q: When?

The contents of the LMU must be a complete conceptualization, even if the answer from which it is derived is not complete. This is not a problem because the memory retrieval unit of QUALM always produces a complete conceptualization, even if the answer generated is a short answer (see Chapter Five). Entering a new concept into the LMU is the last process performed by the memory search unit before the conceptual answer is passed to the generator.

When the next question received by the interpreter is conceptually incomplete, conversational continuity rules are responsible for combining the current question with the LMU concept to form a complete conceptual question. For example, the question 'When?' can be combined in a straightforward manner with the LMU concept of John having left for New York to form a complete conceptual question which asks 'When did John leave for New York?'

3.1.3.3 Pronominal Reference

A slight variation on the basic continuity completion rule is needed when pronominal reference must be resolved in a question. Whenever a pronoun appears in a question, the last concept communicated to the questioner (the LMU) will contain the pronominal referent.

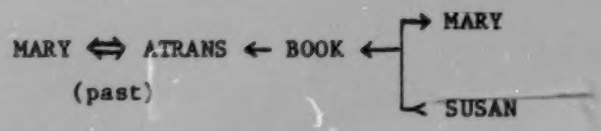
A: Is John still in charge?
B: No, Bill is. (Bill is in charge)
A: Who's he? (who's Bill?)

In this case the pronominal referent for 'he' is easy to find. The last concept communicated (Bill is in charge) contains only one male human in its conceptualization, Bill. So 'he' must refer to Bill.

If there is more than one candidate for pronominal reference among those memory tokens appearing in the last concept, then a pattern match between the question concept and the last concept will determine the correct referent.

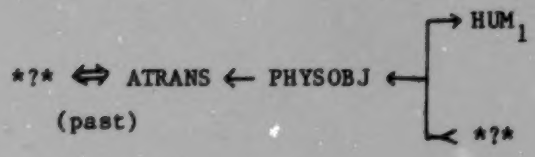
A: Did Mary get the book from Susan?
B: No. (Mary did not get the book from Susan)
A: Who gave it to her? (Who gave the book to Mary?)

In this example the pronominal reference is a little harder. The last concept communicated (Mary did not get the book from Susan) has only one inanimate object, the book. So the referent for 'it' is resolved to be the book. But there are two possible candidates for 'her.' Both Susan and Mary are female humans in the LMU. The correct referent must be found by matching the question concept against the LMU to find which component of the last concept communicated corresponds to the pronoun in question. The conceptual representation for 'Mary did not get the book from Susan' is:



ACTOR = MARY
OBJECT = BOOK
RECIPIENT = MARY

The conceptual representation for 'Who gave it to her?' is:



ACTOR = *?*
OBJECT = PHYSOBJ
RECIPIENT = HUM_1

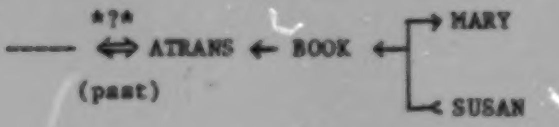
Where HUM has the properties of being human and female. A simple pattern match which recognizes corresponding slots in two conceptualizations and checks property list values can determine that 'her' (the recipient) corresponds to the filler of the recipient slot of the LMU: Mary. Therefore the question is interpreted to mean 'Who gave the book to Mary?

A: Did Susan give the book to Mary?
B: No. (Susan did not give the book to Mary)
A: Was it given to her by Ann? (Was the book given to Mary by Ann?)

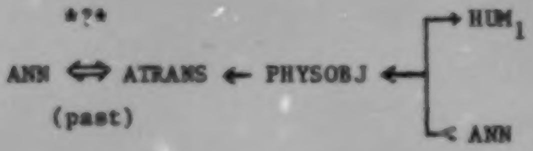
This example is identical to the last one in processing complexity. In Conceptual Dependency, the representation for this dialog is identical[1] to the previous dialog:

A: Did Mary get the book from Susan?
B: No. (Mary did not get the book from Susan)
A: Did Ann give it to her? (Did Ann give the book to Mary?)

Both LMU's 'Did Susan give the book to Mary?' and 'Did Mary get the book from Susan?' are represented by:



Both 'Was it given to her by Ann?' and 'Did Ann give it to her?' are represented conceptually by:



The referent for 'it' can only be the book, and a pattern match against the LMU determines that 'her' (HUM_1) must correspond to Mary.

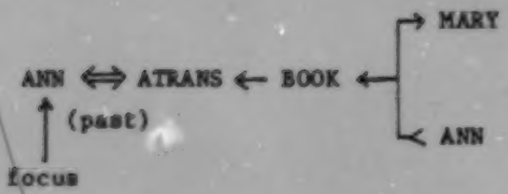
Pronominalization is really a problem for Internalization. Use of the LMU in pronominal reference illustrates how techniques developed for QUALM can spill over into other processing domains.

3.1.3.4 Focus-Based Continuity

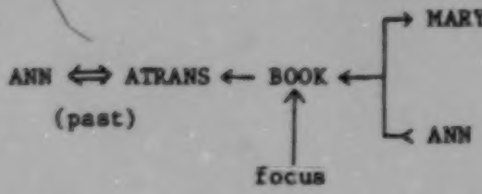
In the last section we said that the same conceptualization represented the questions 'Did Ann give it to her?' and 'Was it given to her by Ann?' This was an oversimplification of the representations, but for the purposes of the pronominalization rule described in 3.1.3.3, it was an adequate statement. There are, however, situations where it is necessary for the conceptual representation to distinguish active and passive constructions. The distinction is accomplished by placing a focus flag on that component of the conceptualization which was emphasized by the sentence structure [Goldman 1974]. Using focus flags, 'Ann gave the book to Mary,' is represented by:

---

[1] This is not exactly true but it is close enough for the point at hand. In the next section (3.1.3.4) we will see how Conceptual Dependency does distinguish active and passive constructions as well as 'Susan gave the book to Mary,' vs. 'Mary got the book from Susan.'


MARY
ANN ⇔ ATRANS ← BOOK
(past)
ANN
focus

'The book was given to Mary by Ann,' is represented by:

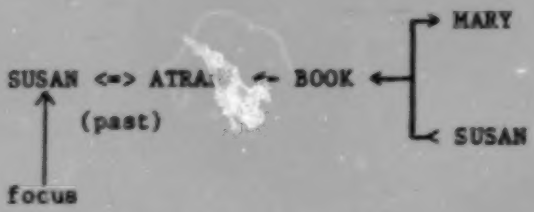

MARY
ANN ⇔ ATRANS ← BOOK
(past)
ANN
focus

This representation of focus in used when concept completion questions must be interpreted in terms of previous dialog.

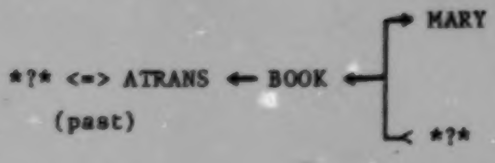
****

DIALOG 1

A: Did Susan give the book to Mary?
B: No. (Susan didn't give the book to Mary)


MARY
SUSAN <=> ATRANS ← BOOK
(past)
SUSAN
focus

A: Who did? (who gave the book to Mary?)

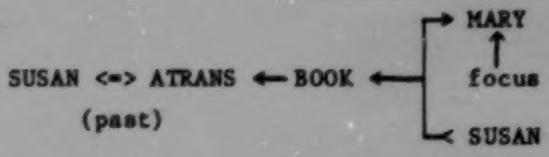

MARY
? <=> ATRANS ← BOOK
(past)
?

****

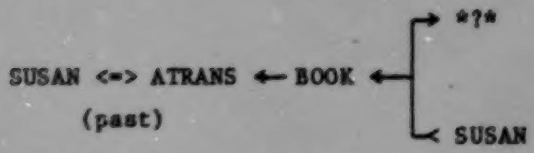
DIALOG 2

A: Was Mary given the book by Susan?

B: No. (Mary wasn't given the book by Susan)


MARY
focus
SUSAN <=> ATRANS ← BOOK
(past)
SUSAN

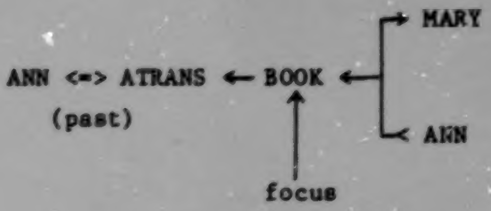
A: Who was? (who was given the book by Susan?)


?
SUSAN <=> ATRANS ← BOOK
(past)
SUSAN

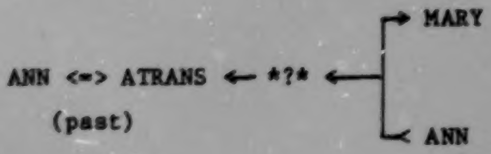
****

DIALOG 3

A: Was the book given to Mary by Ann?
B: No. (the book wasn't given to Mary by Ann)


MARY
ANN <=> ATRANS ← BOOK
(past)
ANN
focus

A: What was? (what was given to Mary by Ann?)


MARY
ANN <=> ATRANS ← *?*
(past)
ANN

In these dialogs the question concept must be inferred on the basis of the LMU. The interrogative pronouns (who, what) must be assigned to a role slot in a complete conceptualization. This conceptualization must be derived from the conversation.

The rule which infers the complete conceptual question correctly uses only the LMU. When the unknown component of an incomplete question does not have an obvious role filler on the basis of a pattern match, then assign the unknown component to the role which received the focus flag, and complete the concept without further alterations. In DIALOG 1, focus fell on the ACTOR slot so 'who' is assigned to the ACTOR slot. In DIALOG 2, focus fell on the TO slot so 'who' is assigned to the TO slot. In DIALOG 3, focus was assigned to the OBJECT slot and so 'what' assumes the OBJECT slot.

3.1.3.5 Knowledge-Based Continuity

More complicated questions which can be interpreted in terms of the basic rule for conversational continuity require general world knowledge to be combined with the LMU to the questioner in order to conceptually complete a question.

A: I want to make a withdrawal.
B: Checking or savings?
(Do you want to withdraw from a checking
account or a savings account?)

In order to understand this question it is necessary to know that 'checking' and 'savings' refer to types of bank accounts. It is also necessary to know that a withdrawal has to do with a bank account, i.e. a withdrawal is a transaction in which money is taken from a bank account. Once these connections are made, the question can be fully interpreted in terms of the LMU.

This kind of knowledge would be found in the conversational script (3.1.1) for bank transactions. So conversational scripts must be accessed along with the LMU in order to complete a particular class of partial questions.

3.2 Context-Independent Inferences

Context-Independent conversion rules may be utilized by the interpreter in any context. These general rules are structured in the same way as contextual inference rules; each one has a set of Criteria and a Target Interpretation. A rule is activated only when its Criteria are satisfied. If all of the Criteria are satisfied, the rule changes the question's conceptual categorization and question concept according to directions specified under the Target Interpretation. These transformations are designed to produce a Target Interpretation which captures an intended question as opposed to a precisely literal question.

Q: Do you do your homework?
A: Almost never.

This question is technically a Verification question which could be answered by a simple yes or no. But the answer given specifies the frequency of the act in question. In order to produce this answer a general interpretive rule had to be applied to the question which re-conceptualizes the question to mean 'How often do you do your homework?'

****

Rule 7: Simple Request Conversion

****

Criteria: (1) Conceptual Categorization = Expectational
(2) Question Concept is of the form:

Y <=> DO_1 TIME (*PRESENT*) MODE (*NEG*)

where *Y* is the person being addressed

Target Interpretation: Conceptual Category ← Request
Question Concept ← *Y* <=> DO_1

*****

Examples:

Rule 7 Applies:

Why don't you listen? ---> Would you listen?

Why don't you come here? ---> Would you come here?

Why don't you pay me now? ---> Would you pay me now?

Rule 7 Does Not Apply:

Why are you paying me now? (wrong conceptual category)

Why didn't you pay me then? (wrong tense)

Why didn't John eat a hamburger? (wrong tense)

****

Rule 8: Frequency Specification Conversion

****

Criteria: (1) Conceptual Categorization = Verification
(2) Question Concept is of the form:
X MANNER (*REPEATEDLY*)
where *X* is a conceptual event

Target Interpretation: Conceptual Category ⇐ Specification
Question Concept ⇐ *X* FREQUENCY (*?*)

*****

Examples:

Rule 8 Applies:

Does it snow in Portland? --->
How often does it snow in Portland?

Do you eat at Clark's? --->
How often do you eat at Clark's?

Do you do your homework? --->
How often do you do your homework?

Rule 8 Does Not Apply:

Did the train arrive this morning? (MANNER = *REPEATEDLY*)

Is John going to New York? (MANNER = *REPEATEDLY*)

The parser is capable of recognizing when an action should have MANNER = REPEATEDLY by accessing scripts in memory. When an action is typically something which occurs again and again, it describes a situational script. Any such situational script will have a descriptive tag which says that this script is normally recurrent.

****

Rule 9: Duration Specification Conversion

****

Criteria: (1) Conceptual Categorization = Verification
(2) Question Concept is of the form:
X TIME (*W*)
where *X* is a conceptual event
and *W* specifies an interval of time

Target Interpretation: Conceptual Category ⇐ Specification
Question Concept ⇐ *X* TIMEDURATION (*?*)

*****

Examples:

Rule 9 Applies:

Have you lived in Italy long? --->
How long have you lived in Italy?

Did it rain this morning? --->
How long did it rain this morning?

Did it hurt for a while? ---> How long did it hurt?

Rule 9 Does Not Apply:

Do you watch football? (would be transformed first by the Frequency Specification Conversion)

Did John go to New York? (does not describe an event over an interval of time)

The last two conversion rules (8 and 9) are optional conversions which do not precisely stay with the immediate meaning of the question. 'Do you beat your wife?' does not have to be understood to mean 'How often do you beat your wife?' These conversions should be sensitive to how talkative the system is: they should only be used if the system wants to be talkative and prolong conversation. If the system is more business-like and less conversational, these conversions should be ignored. Softer conversion rules of this sort are closer to transitional rules of conversation than strict interpretive processing.

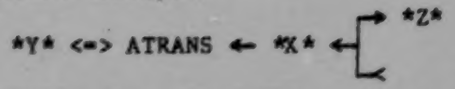
****

Rule 10: Agent Request Conversion

****

Criteria: (1) Conceptual Categorization = Verification
(2) Question Concept is of the form:
X ⟺ POSSBY (*Z*) MODE (*CAN*)
or
Z <=> ATRANS ← *X* MODE (*CAN*)

Target Interpretation: Conceptual Category ← Request
Question Concept ←



where *Y* is the person being addressed

*****

Examples:

Rule 10 Applies:

Can I have a cookie? ---> Would you give me a cookie?

Could we get a collie? ---> Would you get us a collie?

Can Rover have your leftover bone? --->
Would you give Rover your leftover bone?

Rule 10 Does Not Apply:

Can I have swine flu? (is not represented by a POSSBY state;
this is a physical state description)

Does John have a car? (MODE value $\neq$ *CAN*)

Can John see Mary now? (is not a POSSBY or ATRANS)

3.3 Knowledge State Assessment Inferences

The correct interpretation of a question may rely on knowing something about the questioner's knowledge state and goal states. If a mathematician is at a party for university faculty, and someone he has just met asks him:

Q1: What field are you in?

he should interpret Q1 to mean:

Q2: What academic field are you in?

Given this interpretation, he can respond on an appropriate level of description: 'I'm in mathematics.' But if he is at a professional mathematics conference, Q1 should be understood to be asking:

Q3: What specialty field are you in?

Q3 should be answered with a more esoteric description: 'I'm in algebraic topology.' These interpretations are based on assumptions about what the questioner knows and doesn't know.

Knowledge state assessment rules are sensitive to beliefs about the questioner's knowledge state. Each rule has a Criteria which specifies a conceptual categorization and a condition concerning the questioner. If all the Criteria are satisfied, the rule is applied by changing the conceptual categorization and question concept according to the directions under the Target Interpretation. The implementation of these rules is dependent on some sort of memory model for the questioner. How this modeling of the questioner is to be carried out is still a very open problem. A complete model of question answering processes must at some point take into account what the questioner knows and doesn't know, what he wants, and why he asks the questions he does. This problem will be discussed from the viewpoint of memory retrieval in Chapter Eight.

****

Rule 11: Goal Orientation Conversion

****

Criteria: (1) Conceptual Categorization = Concept Completion
(2) The questioner knows the answer

Target Interpretation: Conceptual Category ← Goal Orientation
Question Concept ←
the conceptual answer to the
Concept Completion question

*****

Examples:

Rule 11 Applies:

What are you doing? (To wife packing suitcase)
---> Why are you packing a suitcase?

What have you done? (To a child holding a
torn picture in hand)
---> Why did you tear up the picture?

What are you doing? (To friend throwing the I-Ching)
may or may not be transformed depending
on whether or not the person asking knows
what the I-Ching is

****

Rule 12: Specification Constraint Conversion

****

Criteria: (1) Conceptual Categorization = Specification
(2) The questioner knows certain specifiers

Target Interpretation: Conceptual Category ← Specification
Question Concept ←
same as before but with added
constraints on known specifiers

*****

Examples:

Who is John Dean?
(Don't give his name or sex)

Who is the Secretary of Defense?
(Don't give his title or information which can be inferred from the title - like being a presidential appointee)

A Specification question is one which asks for a feature description of something. e.g. 'Who is the President?' is a Specification question which is best answered by specifying the name of the President. But other feature descriptions could be used, some of which would convey different overtones. If 'Who is the President?' were answered 'Betty Ford's husband,' the answer suggests that you might recognize her name more easily than his. The way in which people answer Specification questions is strongly determined by what they think the questioner knows and doesn't know.

****

Rule 13: Obvious Request Conversion

****

Criteria: (1) Conceptual Categorization = Verification
(2) Question Concept is of the form:

$*Y* <=> DO_1$ TIME (*PRESENT*)

or

$*Y* <=> DO_1$ TIME (*FUTURE*)

where *Y* is the person being addressed, and the questioner is assumed to desire the performance of DO_1

Target Interpretation: Conceptual Category ← Request
Question Concept ← $*Y* <=> DO_1$

*****

Examples:

Are you going to walk the dog?
(is probably a request)

Would you take out the garbage?
(is almost certainly a request)

Would you let your son marry one?
(is not a request)

****

When a question has passed through Inferential Analysis, it has completed the interpretive phase of its processing. Inferences about the intended meaning of the question have been made and the conceptual categorization is final. Processes can now be executed which extract an answer from memory.

For some questions, the memory search will indicate that the conceptul focus of a question must be established before an answer can be formed. In this case further interpretive processes will be invoked after the memory search is initiated. The heuristics involved in focus establishment will be described in Chapter Six. But for other questions, interpretive processes are completed before the memory search is conducted.

Inferential Analysis is based on a theory of human inference processes. Each conversion rule described here represents a cognitive process which people use when they understand questions. These mechanisms are utilized without conscious awareness, both during understanding and generation. Only when someone purposely violates one of these inferences and takes a question 'literally' are we forced to acknowledge that a question was intended to mean something other than what it said explicitly. The processes of inference which people utilize are formalizable as manipulations of conceptual information within Inferential Analysis.

Preface to Memory Searches: Finding an Answer

0. Introduction

Once a question has been sufficiently understood by Inferential Analysis, the memory search can begin to look for an answer. Chapters Four, Five, and Six describe those processes in QUALM involved in finding a conceptual answer which can be passed to the generator. The generator translates conceptual representations into natural language. Generation[1], like parsing, is not part of the Q/A model. The generator which interfaces with QUALM inputs a conceptual answer and translates it into English. In the same way that QUALM can interchange parsers, a generator for any language could be attached to QUALM to produce answers in that language.

The processes of finding and formulating an answer are split up into roughly two basic processes: Content Specification and Memory Search. Content Specification determines what kind of an answer should be produced and how to look for it, while the Memory Search does the work of actually finding information. Content Specification provides direction for the Memory Search. Occasionally the Memory Search finds something which indicates that the question must be analyzed further before the Memory Search can continue. This is the case when the focus of a question must be established. While focus establishment is more properly an aspect of question interpretation, knowledge-based focus heuristics are described here because they are only invoked when the Content Specification predicts a possible need for them and the Memory Search confirms that they are in fact needed. The intuitive division between understanding questions and finding answers becomes difficult to maintain when the memory search invokes heurisitcs which are essentially interpretive.

1. Deciding How to Answer a Question

Once a question has been understood by decomposition into a conceptual categorization and a question concept, it is time to decide how we wish to answer the question. Should the answer be honest? misleading? Should it go into a lot of detail? Or should it provide as minimal a response as possible and still be correct? To a large extent, these issues can only be addressed in terms of a theory of conversation which accounts for why people say the things they do. While we won't attempt to solve problems in conversation here, it is appropriate to design mechanisms in QUALM which will integrate instructions derived from conversational goals into the Q/A processes.

The factors which determine what people say and how they say it are motivational factors concerned with the context and purpose of conversation. If a model of Q/A does not acknowledge the role that these factors must play in the question answering process, it cannot be viewed as a comprehensive model of human question answering. While factors of social interaction are not very well understood at this time, we can still specify how such factors affect and control question answering. That is, QUALM does not know why it says what it does, but it is ready to answer questions talkatively or minimally if a higher motivating process should appear which can tell it what stance to assume. Until a controlling process appears to interface with QUALM, we can arbitrarily set variables within QUALM which will result in different answers to the same questions.

---

[1] While the term 'generation' may sound suggestive of the entire process which finds an answer and produces a natural language response, it is used here in a more restrictive sense. In this thesis generation is a technical term referring to the process which receives a complete conceptual representation and produces a natural language translation of that conceptual information.

In the context of the following story, QUALM can supply the following answers during a question answering session with SAM:

> John went to New York by bus. Then he went to Leone's. The hostess gave John a menu and he ordered lasagna. John ate and asked for a check. When the check came, he discovered that he didn't have any money so he had to wash dishes.

Q1: Where did John go?
A1a: New York.
A1b: John took a bus to New York.

Q2: Did John eat?
A2a: Yes.
A2b: Yes, John ate some lasagna.

Q3: Did the waitress give John a menu?
A3a: No.
A4b: No, the hostess gave John a menu.

Q4: Did John pay the check?
A4a: No.
A4b: No, John discovered that he had no money and so he had to wash dishes.

Q5: Who gave John a menu?
A5a: The hostess.
A5b: The hostess gave John a menu.

Content Specification can be thought of as an interface device which takes information about the general attitude or mood of the system (from any processes outside of QUALM which can specify such things) and integrates this information into the retrieval instructions which produce an appropriate answer. One kind of instruction it gives to the Memory Search incorporates attitudinal variables and is called an Elaboration Option. Another kind of instruction which Content Specification may pass to the Memory Search is a Category Trace Instruction. Category Trace Instructions take into account what kinds of processes were activated during Inferential Analysis in order to understand a question. That is, some answers reflect the interpretive processing which was used to understand them. In Chapter Four the Content Specification Unit is explained and specific Elaboration Options and Category Trace Instructions are described. These specific rules are not intended to be a complete set of all the Elaboration Options and Category Trace Instructions which a general question answerer must have. They are purely representative and intended to illustrate the general idea of Content Specification rather than definitively exhaust it.

2. Carrying Out Orders

Once we know how to search memory for an answer, somebody has to do the actual digging. This is the job of the Memory Search. The Memory Search is defined by a set of default retrieval heuristics. These default heuristics are generally augmented by specific instructions from Content Specification. But if no special guidance is provided by Content Specification, the Memory Search will resort to its standard default processes to produce an answer.

Memory Searches are organized according to three levels of description within a story representation: Script Structures, Planning Structures, and Causal Chain Representations. When a question asks about static properties or features of things, the memory search resorts to checking information stored in memory tokens. The heuristics devised are designed to take advantage of a story representation as much as possible. As such they are intimately connected to the specific features and properties of script and plan-generated memory representations. In fact, the design of these story representations has been altered and extended to accommodate retrieval heuristics at the same time that the retrieval heuristics have been designed to fit the story representations.

For the most part, the information which is needed to produce answers to questions about stories exists within the story representation which was generated at the time of understanding. But there are questions which can only be answered by activating predictive memory processes in conjunction with the story representation. For example, suppose we ask, 'Why didn't John order a hamburger?' in the context of a story where John orders a hot dog. In order to answer this question we must activate processes which can provide information about why John did what he did instead of something else. This information cannot be found in the story representation alone. Special processes which are activated during the memory search when the story representation is not enough are described in detail in Chapter Seven. These situations should be thought of as an exception to the rule: most questions about a story can be answered on the basis of inferences which were made at the time of understanding and stored in the story representation.

In Chapter Five the default retrieval heuristics of QUALM will be described along with brief descriptions of story representations generated by SAM and PAM.

3. Answering Better by Understanding More

There are times when the processes of question interpretation and memory retrieval are not neatly divided. In Chapter Six, processes for answering questions are described where the initial interpretation of a question is followed by a memory search which leads to a need for further analysis of the question, which in turn requires conducting a memory search, after which a final memory search for the answer is completed. The processing flow of control for these questions is outlined in Figure 3. The focus of a question directs attention to one specific conceptual component of the question. Focus can be very important in determining what a question is asking. For example,

Q4: Was it the waitress who gave John a menu?

is a question which directs attention to the waitress. A conceptually equivalent question which does not carry such a strong focus is:

Q5: Did the waitress give John a menu?

To see that focus affects the answers which a question will take, we need only point out that Q5 can be answered:

Q5: Did the waitress give John a menu?
A5: No, the waitress gave Mary a menu.

But the same answer cannot be smoothly offered in response to Q4:

Q4: Was it the waitress who gave John a menu?
A5: No, the waitress gave Mary a menu.

Q4 demands an answer along the lines of:

Yes, the waitress gave John a menu, or
No, somebody else gave John a menu.

Q4 carries a very strong presupposition that John was given a menu by someone; the question is Who? Q5 carries no such presupposition.

Focus in a question can be established many ways and at different points in the process model. One heuristic for focus establishment in QUALM has been implemented for the SAM system and is described in Chapter Six. This is a knowledge-based heuristic. That is, it relies on general world knowledge for the identification of focus rather than syntactic constructions (as in Q4) or intonational devices (which might be used in spoken dialog).

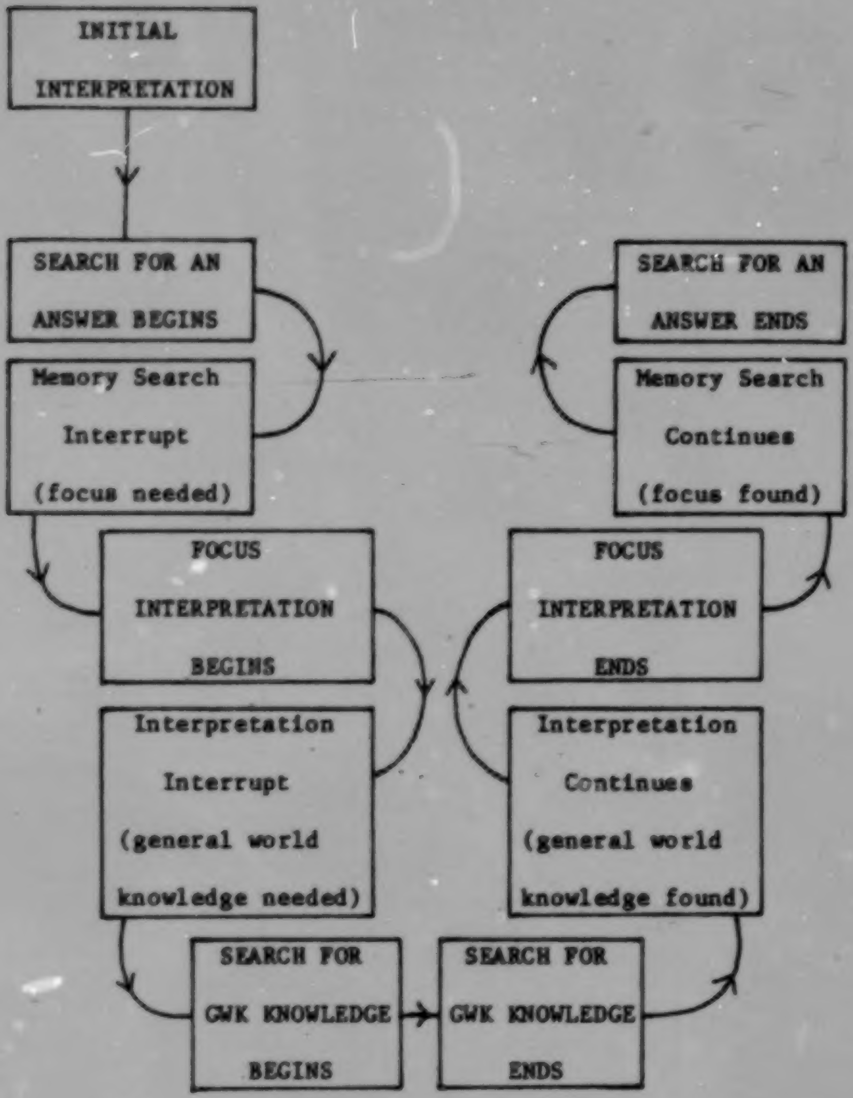

INITIAL INTERPRETATION
SEARCH FOR AN ANSWER BEGINS
Memory Search Interrupt (focus needed)
FOCUS INTERPRETATION BEGINS
Interpretation Interrupt (general world knowledge needed)
SEARCH FOR GWK KNOWLEDGE BEGINS
SEARCH FOR GWK KNOWLEDGE ENDS
Interpretation Continues (general world knowledge found)
FOCUS INTERPRETATION ENDS
Memory Search Continues (focus found)
SEARCH FOR AN ANSWER ENDS

Figure 3

Answering Better by Understanding More:

Knowledge-Based Focus Establishment

4. Generation into English

Once a final conceptual answer has been produced by the Memory Search, all that remains to be done is generation into English (or whatever language is desired). The program which generates from Conceptual Dependency into English is not part of QUALM; it is an independent system which merely interfaces with QUALM. The Generator is based on an original program by Neil Goldman [Goldman 1975] which has since been expanded in terms of vocabulary and finesse. In the course of setting up an interface between QUALM and the Generator, a few minor additions were made to the Generator. The most important of these from a theoretical viewpoint concerns the use of mixed concepts.

There are many times when QUALM's understanding of an answer can be most effectively communicated to the Generator in a mixed format of some connectives and some Conceptual Dependency. For example, in answering a why-question, there is a specific retrieval heuristic which is designed to produce an answer consisting of two causally linked concepts. In all cases the causality between these two concepts is so specific that the two concepts can always be joined together by the connective 'and so.' E.g. 'John discovered that he had no money and so he had to wash dishes,' or 'The waitress told John that they didn't have any hot dogs and so John ordered a hamburger.' Since the retrieval heuristic can reliably predict that any answer obtained via that heuristic can be expressed using those connectives, there is no reason why the Generator should have to hunt around trying to find an appropriate connective.

This is where mixed concepts come to the rescue. The Generator has been extended to accept information structures from QUALM which are part English and part Conceptual Dependency. Mixed formatting is used whenever a retrieval heuristic can predict the connective relationships between concepts. When SAM answers 'John was angry because the hamburger was burnt and so he left,' this answer originally entered the Generator as a mixed concept of three conceptualizations and two English connectives of the form '*X* because *Y* and so *Z*.'

This is the only aspect of QUALM which could conceivably be described as being language dependent. If a Generator for another language were attached to QUALM, the connectives which QUALM inserts in its mixed concepts would have to be altered for the new language. In its current implementations, QUALM uses only the connectives:

1) because
2) and
3) so
4) and so
5) and then

The formatting for mixed concepts in specific retrieval heurisitcs is described as needed in Chapter Five.

In Figure 4 the processes of the Memory Search are outlined. A question ('Did the waitress give John a menu?') enters the Content Specification Unit as a question concept and a conceptual question category. Content Specification consults the system's intentionality and specifies an Elaboration Option to guide the Retrieval Heurisitics. A conceptual answer (representing 'No, the hostess gave John a menu.') is found by the Retrieval Heuristics and passed to the Generator for translation into English.

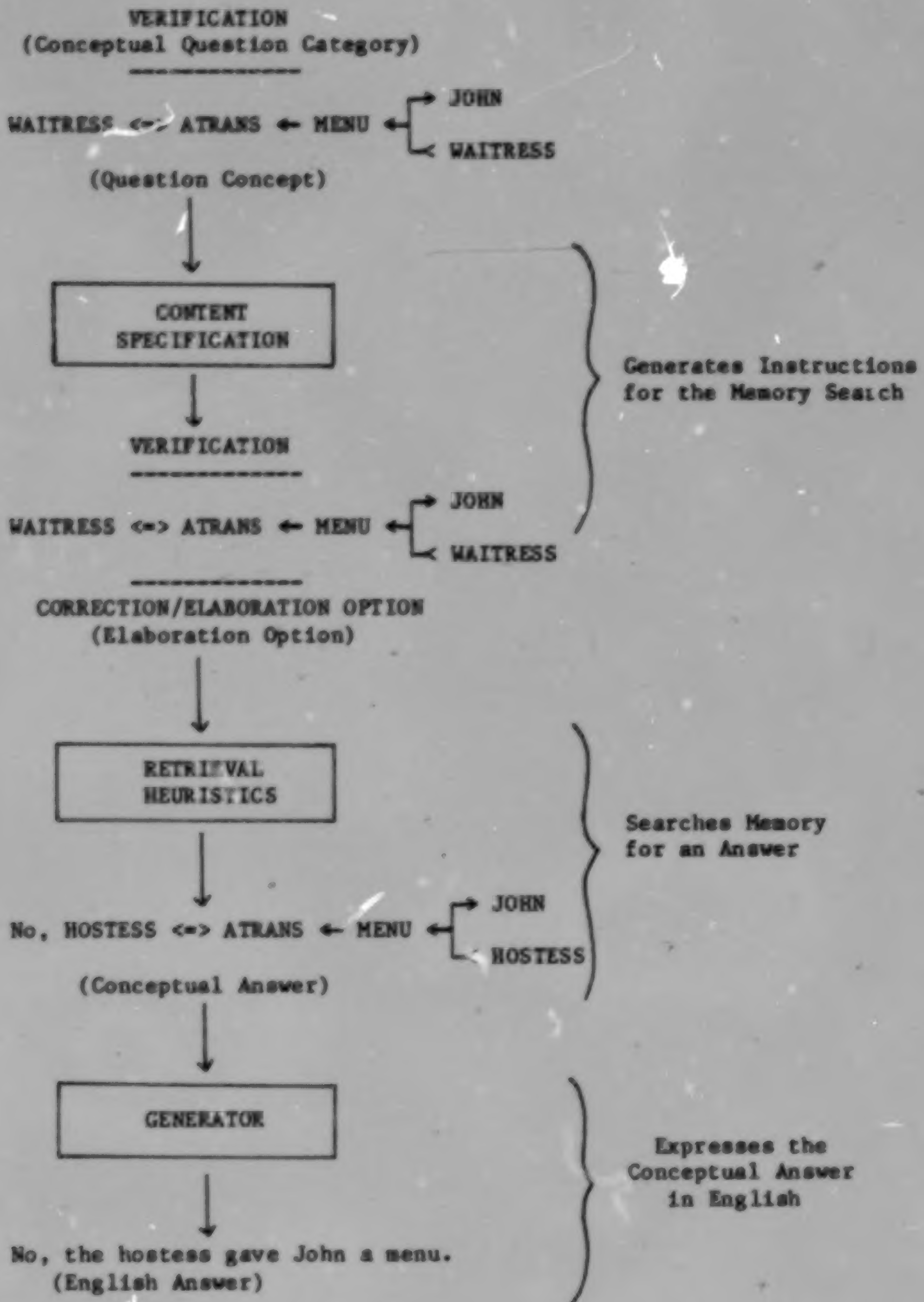

VERIFICATION
(Conceptual Question Category)
WAITRESS <=> ATRANS ← MENU ← JOHN / WAITRESS
(Question Concept)
CONTENT SPECIFICATION
VERIFICATION
WAITRESS <=> ATRANS ← MENU ← JOHN / WAITRESS
CORRECTION/ELABORATION OPTION
(Elaboration Option)
Generates Instructions for the Memory Search
RETRIEVAL HEURISTICS
No, HOSTESS <=> ATRANS ← MENU ← JOHN / HOSTESS
(Conceptual Answer)
Searches Memory for an Answer
GENERATOR
No, the hostess gave John a menu.
(English Answer)
Expresses the Conceptual Answer in English

Figure 4

Memory Search

CHAPTER 4

CONTENT SPECIFICATION

Somewhere within the question answering model must be a process which controls the amount of information which goes into an answer:

Q: Did John apply to Yale?
Aa: No.
Ab: No, his advisors told him he didn't have a chance.
Ac: No, his advisors told him he didn't have a chance so he applied to Harvard instead.

One of the ways that correct answers to a question may vary is in terms of the amount of information communicated. A question answering system must be capable of adjusting responses to yield varying amounts of information.

4.0 Introduction

If a Q/A model always answers the same questions in exactly the same way, it has clearly failed to simulate the way people answer questions. There are two ways in which people vary their answers to questions. On a low level, generational mechanisms can produce answers which are conceptually equivalent, but which are worded differently:

Q1: Why did John order a hamburger?
A1a: The waitress told him they didn't have hot dogs.
A1b: Because he was told by the waitress that they didn't have hot dogs.
A1c: Because the waitress informed him that there were no hot dogs.

On a higher level, answers to questions vary in terms of their conceptual content:

Q2: Did John eat a hot dog?
A2a: No.
A2b: No, the waitress told John they didn't have any hot dogs.
A2c: No, the waitress told John they didn't have any hot dogs and so John ordered a hamburger.

Each of these answers constitutes a reasonable response to Q2, yet they differ in terms of the amount of information they convey. In fact, answers can vary not only in terms of their relative content, but in terms of the kind of content they communicate. For example, if Q2 had been answered 'Yes,' in the context of a story where John didn't eat a hot dog, then the content of this answer would have to be described as dishonest (if we thought the answerer knew better) and wrong (if we thought we knew better).

In this chapter we will look at a mechanism within the memory retrieval part of QUALM which is responsible for determining how much information and what kind of information goes into an answer. This part of QUALM is called the Content Specification Unit. Content Specification takes into account the conceptual category of each question and factors which describe the 'attitudinal' mode of the entire system in order to determine how a question should be answered. A system of descriptive instructions are produced by the Content Specification Unit to instruct and guide memory retrieval processes as they look for an answer.

The primary challenge involved in Content Specification is precisely how these instructions to memory retrieval are formalized. It is not enough to say 'give a minimally correct answer,' or 'bring in everything you can find that's relevant.' The instructions generated by Content Specification must tell the retrieval heuristics exactly how to produce a minimally correct answer and exactly what has to be done to come up with everything that's relevant.

One of the interesting aspects to this problem derives from the fact that initial results from the retrieval processes can affect how the rest of the search should be conducted. That is, the Content Specification Unit cannot give the retrieval heuristics one set of instructions which will work in all cases regardless of what the memory search finds. Content Specification instructions must anticipate in a very general way what kinds of things the retrieval heuristics can produce, and formulate instructions which are sensitive to different retrieval results. For example, a Verification question will require different instructions depending on whether or not the initial search finds the question concept in memory. An affirmative answer will be elaborated differently from a negative answer.

The general system of instructions generated by the Content Specification Unit are described in terms of two basic mechanisms: Elaboration Options and Category Trace Instructions. As soon as a question has been fully interpreted, its conceptual categorization and question concept are passed to the Content Specification Unit. Here the question may be tagged with an Elaboration Option or Category Trace Instruction which will be consulted when retrieval heuristics are executed. If no such special instructions are attached to the question, standard retrieval heuristics (which will be described in Chapter Five) are executed.

But before we can proceed with a description of Elaboration Options and Category Trace Instructions, we must first identify those factors which are ultimately responsible for conceptually different answers. In QUALM, the variables which affect the content of an answer are described as Intentionality factors.

4.1 Intentionality

The factors which affect human question answering are not easy to pin down. Questions are answered differently depending on the context, relationship with the questioner, type of conversational dialog, the purpose of the dialog and everybody's mood. Answers to simple questions are affected by things like:

(1) You know the questioner and can't stand him personally.
(2) You want the questioner to think well of you.
(3) You are taking an oral exam for a PhD.
(4) You are preoccupied with other things.
(5) You want to mislead the questioner.
(6) You find the topic of discussion:
    a) emotionally painful
    b) in poor taste
    c) intellectually stimulating
    d) sexually stimulating
    e) boring
    f) absurd
etc., etc.

The list could go on forever. If we seriously tried to bring all possible variables into our Q/A model, we would quickly find ourselves immersed in some rather deep waters of social psychology. A theory of human conversation (of which Q/A is a part) must eventually touch upon theories of social interaction. But not yet. At this stage it is appropriate to make a few simplistic assumptions.

As far as QUALM is concerned, the attitudinal orientation of a question answerer can be described by a two-dimensional decomposition into (1) Mood, and (2) Reliability. This descriptive decomposition falls far short of specifying a complete attitudinal orientation. The attitudinal factors which it does endeavor to specify are perhaps better described as factors of intentionality. We will view intentionality as that subset of attitudinal factors which are strongly goal-oriented. To be dishonest or extremely informative are attitudinal stances which relate to some ultimate purpose; hence they are intentional. Since we are concerned with designing a mechanism which controls the sort of answers QUALM produces, strongly intentional orientations are reasonable factors to be explored. If it is not already obvious, it should be understood that our notion of Intentionality is a very loose part of QUALM, and absolutely no claims for pyschological validity are being made.

4.1.1 Mood

The first dimension of Intentionality we will call Mood. Variables of Mood can be viewed on a continuum. This continuum is split up into six segments, and the names assigned to each segment are meant to be purely suggestive:

(1) Talkative
(2) Cooperative
(3) Minimally Responsive

(4) Uncooperative
(5) Rude
(6) Condescending/Sarcastic

These attitudinal stances are partially ordered in terms of which Moods imply which other Moods. To be talkative implies cooperativeness, and being condescending, sarcastic, or rude implies being uncooperative. But not all Mood relationships are so clear. There are contexts in which minimal responsiveness can be construed as rudeness (say at a casual party) while sarcasm tends to be rude only when the content of a sarcastic remark is insulting.

4.1.2 Reliabilty

The second dimension of Intentionality is Reliability. There are only two orientations which the Reliability factor can assume:

(1) Honest
(2) Deceptive

If the system is operating with honest reliabilty, its answers to questions will be correct to the best of its knowledge. If the system doesn't know the answer to a question it will say so. When the system is operating with deceptive reliability, answers to questions may be correct and they may be incorrect. The system may give an answer which it knows to be incorrect, it could say it doesn't know the answer when it does, it could guess at answers, or give the right answer. Any of the six Moods can assume either an honest or deceptive Reliability.

4.1.3 Intentionality and Answers

Roughly speaking, the Mood of the system controls the form of the answer while the Reliability of the system controls the content of the answer. As soon as a question has been assigned a final interpretation, each Elaboration Option and Category Trace Instruction checks the Mood and Reliability factors of the system in order to see if special retrieval instructions and answer formation instructions should be attached to that question before retrieval heuristics are executed. Each set of instructions is constrained by specific Moods and Reliabilities which determine whether or not those instructions should be executed.

It is much more difficult to implement deceptive reliability than it is to implement honest reliability since deception in conversation is motivated by various conversational goals; people have reasons for speaking deceptively and there are more ways for an answer to be wrong than right. The actual implementation of QUALM has therefore been limited to a system operating with honest reliability. Until we have a theory of conversation we cannot hope to model deceptive answers in any theoretically sound way.

The system of Intentionality proposed here is adequate for a Q/A model. It would not be adequate in a general model of interactive dialog. Question answering by itself is a largely passive process;

everything is done in response to the questioner. But fully interactive dialogs which maintain mixed initiatives require active goal orientations.

A theory of mixed initiative dialog must (1) identify what kinds of goals people have in conversation, and (2) describe the mechanisms which manifest these goals in converstation. No one in the field of AI has proposed a system of general conversational goals at this time. But many people have worked on specific mechanisms which assume one particular goal and converse accordingly [Bobrow et al. 1976, Collins 1976, Shortliffe 1974]. Of these efforts, the most interesting one from the point of view of conversational theory is the work by Collins on Socratic dialogs. Collins analyzed transcripts of Socratic dialogs in an effort to extract general principles which could be incorporated in a teaching program. His analysis produced 23 goal-oriented strategies which are used according to a hierarchy of goals and subgoals. These strategies motivate the teacher in a Socratic conversation.

To a large extent, a Q/A model can be devised without reference to theories of conversation. We will discuss more carefully where Q/A becomes unavoidably tangled up with rules of conversation in Chapter Ten.

4.2 Elaboration Options

Elaboration Options provide the system with an ability to elaborate its responses to questions. Elaboration Options are summoned for a question only if the Intentionality of the system indicates that an elaboration is appropriate. If the Intentionality 'wills' an elaboration, an appropriate Elaboration Option is given to the question immediately after interpretation. This option is then activated during memory retrieval only if an initial memory search produces information which meets a set of criteria specified by the option.

An Elaboration Option consists of four parts:

(1) Intentionality Threshold (IT)
(2) Question Criterion (QC)
(3) Initial Answer Criterion (AC)
(4) Elaboration Instructions (EI)

The Intentionality Threshold specifies what sort of intentionality the system must be running under in order for the Elaboration Option to be assigned to the question. The Question Criterion describes what conceptual category the question must have in order for it to receive the Elaboration Option[1]. If either the intentionality of the system or the conceptual category of the question fail to meet the specifications of the Intentionality Threshold and the Question

---

[1] In one case (the Inquiry Explanation Option) a MODE specification is needed in addition to the conceptual category.

Criterion, then the Elaboration Option is not tagged to the question. The Initial Answer Criterion specifies the type of conceptual answer which the memory search must initially return in order for the Elaboration Option to be executed. And the Elaboration Instructions specify exactly how an elaboration is to be extracted from memory and integrated into the conceptual answer.

In the following Elaboration Options, the Answer Criterion and Elaboration Instructions will be described non-technically. In an actual implementation of these Options, the Answer Criterion would specify a test to be applied to the initial answer, and the Elaboration Instructions would specify a function designed to retrieve information from memory and integrate that new information with the initial answer for a final conceptual answer. Of the Options described here, only the first four, the Verification, Short Answer, Single Word, and Correction/Explanation Options, have actually been implemented to run in SAM and PAM. The processes which carry out the Elaboration Instructions for the Correction/Explanation Option will be described in Chapter Six.

****

Rule 1: Verification Option

****

IT: Talkative
QC: Verification
AC: initial answer is Yes
EI: final conceptual answer is 'Yes, *X*'
    where *X* is the conceptualization found
    in the story representation which matches
    the question concept

Examples:

Did John go to New York?
Yes, John went to New York by bus.

Did John eat?
Yes, John ate lasagna.

Did someone pick John's pocket?
Yes, a thief picked John's pocket.

Did John pay the check?
Yes, John paid the bill.

Answers produced using this option often appear to be volunteering more information than the question specifically asks for. When the question 'Did John eat?' is answered with 'John ate lasagna,' we are told not only that John ate but what he ate as well. This happens whenever the question concept matches a concept in the story which encodes more information than the question concept contained. The matching heuristic does not require an exact match between the question concept and a conceptualization in the story representation. In order for a match to occur, a conceptualization must have everything in the question concept but it can also include more than is in the question concept.

****

Rule 2: Short Answer Option

****

IT: Minimally Responsive, Uncooperative, or Rude
QC: Verification
AC: initial answer is Yes (or No)
EI: final conceptual answer is Yes (or No)

Examples:

Did John eat lasagna?
No.

Did the waitress bring John a menu?
No.

Did the hostess bring John a menu?
Yes.

Did John go to New York?
Yes.

The Initial Answer Criteria for this option is superfluous since it does not weed out any questions. This option and the next one are perhaps better described as Anti-Elaboration Options since the answers they produce are minimal. The technical realization of Elaboration Options is a control structure for retrieval heuristics. This control can result in a non-elaboration as well as an elaboration.

****

Rule 3: Single Word Option

****

IT: Minimally Responsive, Uncooperative, or Rude
QC: Concept Completion
AC: initial answer is found
EI: final conceptual answer is that conceptual
    component which matched the unknown component
    of the question concept.

Examples:

Who gave John a menu?
The hostess.

What did John eat?
Lasagna.

Where did John go?
New York.

Who went to Leone's?
John.

The other way that a Concept Completion question can be answered is by generating back the entire conceptualization (Where did John go? - John went to New York). In the implementation of QUALM used for SAM and PAM, the default retrieval heuristics for Concept Completion questions produce a long answer and the Single Word Option must be activated to produce a short answer. It could have been the other way around just as easily. The Single Word Option is actually more of a suppression option than an elaboration option in the sense of extending an answer.

****

Rule 4: Correction/Explanation Option

****

IT: Cooperative or Talkative
QC: Verification
AC: initial answer is No
EI: (1) Use the Focus Heuristic[2]
        to make a Concept Completion question
    (2) Answer the Concept Completion question

(3) If an answer is found, append this conceptual answer to the answer No

(4) If no answer is found for the Concept Completion question, make an Expectational question[3], answer it, and append the conceptual answer to the answer No

Examples:

Did the waitress give John a menu?
No, the hostess gave John a menu.
(answers the Concept Completion question:
Who gave John a menu?)

Could John pay the check?
No, John had no money.
(answers the Expectational question:
Why couldn't John pay the check?)

Did the waitress serve John a hot dog?
No, the waitress gave John a hamburger.
(answers the Concept Completion question:
What did the waitress give John?)

Did John eat the hamburger?
No, John was angry because the hamburger
was burnt and so he left.
(answers the Expectational question:
Why didn't John eat the hamburger?)

The first part of the Correction/Explanation Instruction attempts to correct the question concept. If the question concept to be verified is almost right except for one conceptual component (say the actor or the tense), an appropriate elaboration will make the necessary correction. The second part of the Correction/Explanation Instruction is used when no correction is possible. In this case the concept in question could have happened but didn't. The Correction/Explanation Option then endeavors to explain what interfered or what was responsible for a different turn of events.

---

[2] The Focus Heuristic will be described in 6.3.

[3] This means to enter memory retrieval with the question category = Expectational instead of Verification. The question concept does not need to be changed.

****

Rule 5: Inquiry Explanation Option

****

IT: Talkative or Cooperative
QC: Verification with Code (*NEG*)
AC: initial answer is Yes
EI: find and explain apparent inconsistency

Examples:

Weren't you going to California this week?
Yes, but my husband got sick so I postponed it.

Aren't you a member here?
Yes, but I don't have my card with me.

Can't you drive to work?
Yes, but I'd rather not unless I have to.

The questioner thinks that the non-negated concept in question is true (you were going to California, you are a member, you can drive to work), but he finds it surprising or inconsistent with other information. He is looking for some explanation which will allow him to integrate the question concept more satisfactorily. It is up to the person being addressed to supply the explanatory information which is needed. There are some general rules for finding such explanations.

If the question concept describes an act, the explanation can be found by answering an Expectational question.

Aren't you going to New York? --->
Why aren't you going to New York?
(I couldn't get my car started.)

Isn't he taking the job? --->
Why isn't he taking the job?
(He decided to hunt around some more.)

If the question concept describes a state, the explanation is harder to find. If the question asks about an enablement condition, it can be answered by answering an Enablement question. But finding which Enablement question should be generated is more than a trivial manipulation of the question concept.

Aren't you a member here? --->
Can you get in here?
(I don't have my card with me.)

Aren't Porsches terribly expensive? --->
How can you afford a Porsche?
(My parents gave it to me.)

The correct interpretation of questions like these require additional inferences about the questioner's knowledge state, beliefs, and expectations.

Still other states lend themselves to elaboration strategies which are much more conversational:

Isn't British Columbia beautiful?
(Yes, and it's a great place to catch salmon.)

****

Rule 6: Request Explanation Option

****

IT: Talkative, Cooperative, or Minimally Responsive
QC: Request
AC: initial answer is No
EI: make an Expectational question and append the answer to No

Examples:

Would you pass the salt?
No, I can't reach it.

Can we balance the checkbook now?
No, I have a headache.

Are you going to walk the dog now?
No, I have to make a phone call first.

When a request is denied it is a standard civility to offer an explanation for the denial.

****

Rule 7: Delay Specification Option

****

IT: Talkative, Cooperative, or Minimally Responsive
QC: Request
AC: answer is Yes but with expected delay
EI: specify the expected time lapse
and append this to Yes

Examples:

Would you pass the salt?
Yes, as soon as I put this down.

Will you take me to a ball game?
Yes, maybe next week.

Are you going to take out the garbage?
Yes, in a minute.

When someone is willing to fulfill a request but cannot do it immediately, the person making the request must understand that it will be taken care of eventually.

****

Rule 8: Condition Specification Option

****

IT: Talkative, Cooperative, or Minimally Responsive
QC: Request
AC: the answer is yes if certain conditionals are met
EI: specify the condition and append to Yes

Will you go to the store?
Yes, if I can get the car started.

Can we get a new car?
Yes, as soon as I get a raise.

Will you give me the furniture, the house, and the car?
Yes, if you'll leave me alone.

When someone agrees to a Request they are involving themselves in a social interaction which may have many consequences beyond the simple performance of the act. If John asks Mary to do his laundry for him all the time and Mary agrees gratis, her agreement indicates a great deal about their relationship. Conditionals are often used on these higher levels of social interaction to clarify relationships and goals. If Mary agrees to do his laundry only if he marries her, she has used the request as a vehicle for higher levels of communication.

****

Rule 9: Inference Anticipation Option

****

IT: Talkative
QC: Verification
AC: Answer is Yes but with misleading inferences*
EI: correct misleading inferences after Yes

Examples:

Do you have a bicycle?
Yes, but I never ride it.

Aren't those terribly expensive?
Yes, but this one was given to me.

Is your brother applying for a scholarship?
Yes, but he doesn't have a chance.

This is a way to prolong or contribute to conversation. When an elaboration describes something running counter to normal expectations, the elaboration is pointing out something which is at least minimally interesting and which may be pursued for further explication.

****

The Elaboration Options described thus far have all used very simple tests for their Initial Answer Criteria. These tests were all independent of the specific retrieval heuristics which found the intitial answer. Another type of Elaboration Option can be designed to rely on the specific retrieval heuristics which are successful in finding an answer. It is difficult to give examples of these now

---

[*] Before we can formalize a notion of 'misleading inferences' we must first be able to model the inferences which someone is liable to make when they hear an answer to a question. This problem will be discussed further in section 4.2.1.

before retrieval heuristics have been discussed. But the next Elaboration Option is an example of a retrieval-oriented mechanism for content specification.

The basic idea involves replacing the Initial Answer Criteria with Retrieval Criteria (RC) specifying a particular retrieval heuristic. If this heuristic succeeds in finding an answer, then the Option is executed. The following rule is an example of an Elaboration Option with Retrieval Criteria. The retrieval mechanism specified will be described in section 5.4.1.2.

****

Rule 10: Mental State Description Option

****

```
IT: Talkative
QC: Causal Consequent
RC: Interference/Resolution Search
EI: If the actor of the Resolution concept undergoes
    a mental state change at some point between the
    Interference concept and the Resolution concept,
    return a mixed format answer of the form:
       '[M] because [I] and so [R]'
    where
       [M] is the mental state change
       [I] is the Interference concept
       [R] is the Resolution concept
```

This Elaboration Option is responsible for an answer like:

> Q: What happened when the waiter served the hamburger?
> A: John became angry because the hamburger was burnt and so he left.

Whereas the default retrieval heuristics for an Interference/Resolution Search would have produced the answer:

> A: The hamburger was burnt and so John left.

4.2.1 Preventative Inference Simulation

Often the Answer Criterion of an Elaboration Option can be satisfied by a simple examination of the initial answer. It may be enough to simply see if the initial response is Yes or No. But some Answer Criteria require additional memory processing. For example, the Answer Criteria for the Inference Anticipation Option requires some very involved memory processes. The Inference Anticipation Option is responsible for elaborations like:

Q1: Do you like Indian food?
A1: Yes, but New Haven has no good Indian restaurants.

This elaboration is generated by a mechanism which checks the LMU[4] after the initial response 'Yes' is given. The LMU resulting from the answer 'Yes' in this case contains the conceptualization for 'I enjoy eating Indian food.' The elaboration option which is triggered in this situation must look something like:

```
IF (1) the LMU describes an act with
       MANNER value = HABITUAL, and
   (2) an enabling condition for that act is not satisfied

THEN augment the initial response with a description
     of the missing enablement.
```

This mechanism is responsible for exchanges like:

Q2: Does John shop at Bloomingdale's?
A2: Yes, but he can't afford it.

Q3: Do you usually drive to work?
A3: Yes, but right now my car isn't running.

In order to simulate the inference processes which are used by speakers in dialog, the conversational context is critical in guiding the flow of inference. Actual dialog is often instrumental to the achievement of common or individual goals:

Q4: Do you want to play tennis?
A4: Sure.
Q5: Do you have a racket?
A5: No, but I can borrow one.

In this conversation both participants share a common goal; they want to play a game of tennis. Given this goal orientation, there is an obvious purpose behind Q5. The person asking Q5 wants to know the status of the enabling condition of having a racket. When the purpose of Q5 is understood in this way, it is easy to see how a simple 'No' answer to Q5 would result in a misleading inference. The elaboration in A5 addresses itself to the goal implicit in the last question. In this way a misleading inference (that they can't play because they need a racket) is prevented by the elaboration. Without knowledge of the conversational context and instrumentality, an elaboration might be constructed which prevents a totally irrelevant inference:

Q6: Do you want to play tennis?
A6: Sure.
Q7: Do you have a racket?
A7: No, but I used to have one.

---

[4] The LMU was introduced in section 3.1.3.2

This elaboration prevents us from inferring that the respondent never had a racket. This assurance seems out of place because we cannot relate it to the purpose of the dialog. Appropriate elaborations which prevent undesireable inferences are motivated by conversational goals. Inference simulation must be directed by underlying conversation and so it tends to be a problem closer to conversation theory than question answering per se.

4.3 Category Trace Instructions

Category Trace Instructions are responsible for producing answers which reflect the interpretive processing a question undergoes. There are many situations where appropriate answers to questions actually indicate that the question underwent successive interpretations. For example, in Chapter Three we described a Frequency Specification Conversion within Context-Independent Inferential Analysis. This transformation was responsible for recognizing when Verification questions should be understood to be asking how often an event takes place:

Q1: Do you eat out very often?
A1a: About twice a week.

Q2: Do you see each other?
A2a: Only during holidays.

Q3: Is there much rain?
A3a: Only in December.

Once these questions have been ultimately interpreted as Specification questions, the memory search can only recognize them as such and answer them by specifying a frequency. Answers to Verification questions which have been reinterpreted as Specification questions are different from answers to questions which were initially interpreted as Specification questions. Each of Q1-3 can be answered with an intitial response of yes or no:

Q1: Do you eat out very often?
A1b: Yes, about twice a week.

Q2: Do you see each other?
A2b: Yes, but only during holidays.

Q3: Is there much rain?
A3b: No, only in December.

But if Q1-3 were rephrased as questions which would be initially interpreted as Specification questions, these prefaces of yes and no would be out of place:

Q4: How often do you eat out?
A1b: Yes, about twice a week.

Q5: How often do you see each other?
A2b: Yes, but only during holidays.

Q6: How often does it rain?
A3b: No, only in December.

Q4-6 are conceptually equivalent to Q1-3 but they cannot be answered in the same ways that Q1-3 can be answered. Q4-6 can be answered with A1a-3a but not with A1b-3b. The answers A1b-3b are appropriate for Q1-Q3 because they reflect the interpretive processing which Q1-3 undergo. Since Q4-6 do not require the same reinterpretation as Q1-3, answers which reflect reinterpretive processing are inappropriate for these questions.

Category Trace Instructions are designed to construct answers which reflect the interpretive processing of a question. The only information which needs to be recorded about a question's interpretive processing is a history of that question's conceptual categorizations as it passes through Inferential Analysis. A trace of interpretive categorizations is easy to maintain in a simple list. We will call this list the Category Trace. Each of the questions Q1-3 have a Category Trace = (Verification, Specification/(Frequency)). Questions Q4-Q6 have a simple Category Trace = (Specification/(Frequency)).

Category Trace Instructions are dependent on two factors: the Intentionality of the system and the Category Trace of the question. Each Category Trace Instruction has three parts:

(1) Intentionality Threshold (IT)
(2) Trace Criterion (TC)
(3) Instruction Execution (IE)

The Intentionality Threshold describes under what Intentionality factors the Trace Instructions will be appropriate. The Trace Criterion specifies what kind of Category Trace a question must have in order to qualify for the Trace Instructions. If either the Intentionality Threshold or the Trace Criterion specifications fail to be met, the Trace Instructions will not be applied to the question. The Instruction Execution describes the processes which will derive a final conceptual answer in the event that the Trace Instructions are executed.

****

Rule 11: Verifying Frequency Instructions

****

IT: Talkative, Cooperative, Minimally Responsive
TC: Verification/Specification (Frequency)
IE: yes or no, followed by frequency specification

Examples:

Do you eat out much?
No, very rarely.

Did John study in college?
Yes, night and day.

Does it snow in Portland?
Yes, about once every year or two.

Does the computer ever crash?
No, never.

****

Rule 12: Verifying Duration Instructions

****

IT: Talkative, Cooperative, Minimally Responsive
TC: Verification/Specification(Duration)
IE: yes or no, followed by duration specification

Examples:

Was Nixon in office long?
Yes, about six years.

Did John stand there for more than an hour?
No, he was there about forty minutes.

Have you lived in Italy long?
No, only a month.

Are you leaving soon?
Yes, as soon as I get my papers.

4.3.1 Category Trace Instructions vs. Elaboration Options

There is a very fundamental difference in answers which are the result of a Category Trace Instructions and answers which are derived from Elaboration Options. By looking at the answers alone, the distinction is not apparent. Both of the answers:

> No, the hostess gave John a menu.
> Yes, about once every year or two.

appear to assume the same form. They are both Yes or No answers followed by an elaboration of some sort. The difference between a compound answer derived from a Category Trace Instruction and one derived from an Elaboration Option has to do with the order in which conceptual components of the answer are found during the memory search. When a question goes into the memory search with an Elaboration Option, the initial answer can be handed to the generator before the elaboration is found. In fact, Elaboration Options are designed so that the initial answer is a prerequisite condition for the application of an Elaboration Option. If asked,

Q7: Did McGovern win in 1972?

it is possible to generate the response 'No' as soon as the question concept is found to be false. An elaboration (Nixon won in 1972) can then be found and generated after the initial response 'No.'

With questions which are answered by a Category Trace Instruction, the initial answer (Yes or No) depends on the elaboration. The last part of the answer must be found before the first part of the answer can be generated. In order to answer 'No' in response to 'Does John go to the opera?' one must know how often John goes to the opera. In order to answer 'Yes' to 'Has John been to Europe very often?' one must know how often John has to been to Europe. But to answer No to 'Did McGovern win in 1972?' it is not necessary to know that Nixon won in 1972.

Category Trace Instructions also differ from Elaboration Options in terms of the system Intentionality required to activate them. Category Trace constructions are almost mandatory: the system must be operating in an uncooperstive or rude Intentionality in order for a Category Trace construction to be suppressed. The Intentionality Threshold for many Elaboration Options is much higher. Some Elaboration Options are triggered only if the system is talkative.

4.3.2 Hardly Ever Is Never Very Often

There are some subtle difficulties involved in the formation of compound Category Trace answers. The problems arise in finding the approriate initial response. Consider how many ways you could answer:

Q8: Do you go to New York very often?

A8a: No, only on Sundays.
(once a week is not very often)

A8b: Yes, every Sunday.
(once a week is very often)

A8c: No, hardly ever.
(hardly ever can never be very often -
if it could, we could say 'Yes, hardly ever.')

A8d: No, never.
(never is not very often -
if it were we could say 'Yes, never.')

A8e: Yes, whenever I can.
('whenever I can' must be very often)

A8f: No, only when I have to.
('only when I have to' can't be very often)

A8g: Yes, whenever I have to.
('whenever I have to' must be very often)

A8h: No, but whenever I can.
('whenever I can' must not be very often)

A8i: Yes, but only when I have to.
('only when I have to' must be very often)

A8j: No, but whenever I have to.
('whenever I have to' is not very often)

The choice of an initial answer here is dependent on whatever cut off point is selected for 'very often.' A8e and A8h show how 'whenever I can' can take either initial response, but there is something about this answer which makes it more consistent with a Yes than a No. So A8h must be constructed with the connective 'but' to signal a violation of expectations.

Do you go to New York very often?
No, whenever I can.

sounds very strange. While

Do you go to New York very often?
No, but whenever I can.

seems perfectly fine. The same thing happens with A8f, A8i, A8g, and A8j. 'Only when' answers are consistent with 'No' (A8f), and 'whenever' answers are consistent with 'Yes' (A8g). If 'only when' answers are combined with 'Yes,' or 'whenever' answers are combined with 'No,' a but-construction is needed to counter the violated expectation.

4.4 Elaborating Elaboration Options

It may seem odd that an entire chapter on Content Specification has been presented which seems to totally ignore the problem of knowledge state assessment. That is, the content of an answer should address gaps in the questioner's knowledge state. If the mechanisms which control memory retrieval are not motivated by a model of what the questioner does and doesn't know, then where will knowledge state assessment fit in?

If we had a theory of knowledge state assessment, it would interface with QUALM during Content Specification. Each Elaboration Option and Category Trace Instruction would specify knowledge state assessment criteria (KSAC) which would apply tests at the time of memory retrieval to make sure the elaboration is not telling the questioner something he already knows.

Knowledge state assessment will be discussed in Chapters Eight and Ten. It is an area which needs attention. A theory of knowledge state assessment would be a significant contribution to work in question answering and theories of conversation. When such a theory is sufficiently developed, QUALM will be ready to interface with knowledge state models during Inferential Analysis and Content Specification by refining the Conversion Rules and Elaboration Options proposed here. In the same way that Content Specification mechanisms are sensitive to intentionality factors without knowing what a theory of intentionality should look like, the interpretive mechansisms and memory retrieval controls proposed for QUALM have been designed to accommodate input from a knowledge state model, even if we don't know exactly what that model will look like.

CHAPTER 5

SEARCHING MEMORY

Before information can be extracted from memory, it must be found. If memory organization and retrieval heuristics are designed with care, searching processes should be able to zero in on the desired information without having to sift through everything in memory. If a system is trying to answer the question:

Q: Is New Haven in Connecticut?

It should not begin by looking at:

Lassie is a collie.
I burnt an English muffin this morning.
Jack Benny is dead.
The integers under addition form an abelian group.

The time required for a search which examines irrelevant information without any sense of direction will grow linearly with the amount of information in memory. That is, the more you know, the longer it will take to remember something. This does not sound like a very promising theory of memory. The processes which search memory must know where they're going before they begin; it is not sufficient to flit around like a blind bat.

5.0 Introduction

The memory search heuristics invoked for a question are determined by its conceptual question category. There are three levels of story representation where answers to questions may be found:

(1) the causal chain representation
(2) script structures
(3) planning structures

Each of these levels represents a different level of understanding and detail. Not all stories have representations on each level. SAM generates story representations which have a causal chain and script structure, while PAM generates story representations made up of a causal chain and plan-based structures. But in a system which accesses both scripts and plans during its understanding, story representations will entail all three descriptive levels. The retrieval heuristics described here were designed with such a system in mind.

Once the Inferential Analysis has settled on a conceptual question category, and Content Specification has determined a search strategy, the memory search can begin to look for an answer. Since each question category requires different processing by the memory search, the description of retrieval heuristics will be organized according to conceptual question categories. But before we describe techniques specific to each conceptual question category, we must first mention a fundamental process which is used for most of the categories: the matching search.

Very often the first order of business when searching any level of story representation is to find a conceptualization which matches the question concept. This process is called the matching search. Roughly speaking, the matching search looks for a conceptualization which has everything that the question concept has, and which may contain additional things not found in the question concept. So if the question concept represents 'John took a bus,' (as would be the case for the question 'Why did John take a bus?'), and the script summary for the bus trip represents 'John went to New York by bus,' the matching search will accept this script summary as a match. The Conceptual Dependency representations for these two concepts differ only in that 'John went to New York by bus,' specifies a destination for the PTRANS while 'John took a bus,' contains no destination. The term 'answer key' is used to refer to that conceptualization from the story representation which matches the question concept when the matching search is successful.

The heuristics described here do not represent a complete or definitive set of search strategies. This is more of a beginning, representing those heuristics which have been implemented in a computer program and which appear to be adequate for large classes of questions in the context of story understanding. Unless there is an indication to the contrary, all retrieval heuristics described in this chapter have been implemented in the versions of QUALM which are used by SAM and PAM.

5.1 Causal Antecedent

A causal antecedent question may find its answer in either the causal chain representation, script structures, or planning structures. In each case, the first step is to find an answer key.

5.1.1 Script Structure Retrieval Heuristics

The matching search for Causal Antecedent questions begins with a search of script structures. Each script instantiation which is referenced by the script structure representation has a pointer to a single conceptualization which is called the script summary. For example, in the Leone's story, the script instantiation of the bus ride to New York has a script summary which represents John having gone to New York by bus. The script instantiation of his restaurant episode has a script summary representing John having eaten lasagna at

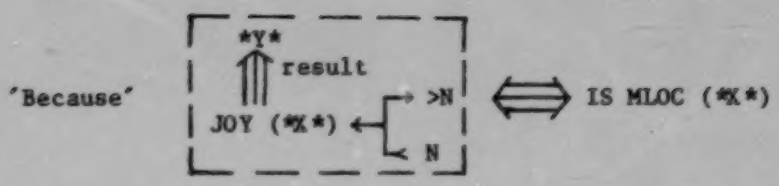
Leone's. For a more detailed discussion of script summaries, what they are and how they are created for a story representation, see [Cullingford 1976, 1977]. Searches of the script structure for Causal Antecedent questions access only these script summaries when looking for an answer key.

If the matching search succeeds at the script structure level, the question is answered according to script-specific retrieval heuristics. That is, each script specifies a set of retrieval heuristics which are organized according to conceptual question categories. These heuristics apply only to questions which locate an answer key in the script structure search. It is probably more illuminating to discuss how a few specific questions are handled by script-specific heuristics, rather than describe all the heuristics of all currently implemented scripts.

Suppose we asked 'Why did John go to New York?' in the context of the Leone's story. This is a Causal Antecedent question with a question concept representing 'John went to New York.' The matching search accesses the script structure of the story representation and determines that the top level script under which the entire story is embedded is the trip script. It then examines the script summaries from the trip script instantiation and all script instantiations embedded within the trip script. When it checks the script summary for the bus script instantiation (John went to New York by bus) it finds the answer key.

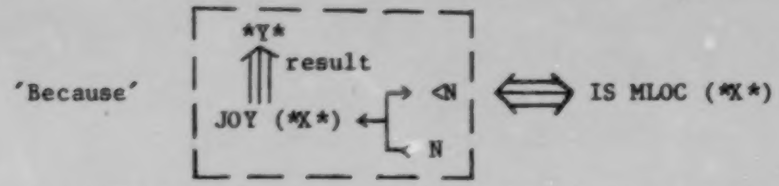
Since the matching search succeeded while examining a trip script instantiation, a special heuristic specific to that script is now invoked. The heuristic first determines whether the answer key was a summary of a script instantiation which was part of the (1) going, (2) destination, or (3) returning leg of the trip script. An answer is then produced which is dependent on in which leg of the trip script the matched script summary was found.

If the answer key was found in the 'going' leg of the script, as was the case when we asked 'Why did John go to New York?' then the answer is produced by concatenating the script summaries of all scripts instantiated in the destination leg of the trip. In the Leone's story there is only one script instantiated in the destination leg of the trip script, the restaurant script. So the conceptual answer to 'Why did John go to Leone's?' involves the script summary of the restaurant script instantiation: 'John went to Leone's.' But before the final conceptual answer can be sent to the generator, a small manipulation must be performed on the script summary so that its statement relates back to the question. In this case some indication of an enabling relationship is appropriate. This can be achieved by generating a sentence of the form: 'Because X wanted to Y,' where X is the actor of the answer key and Y is the conceptual action of the script summary (or summaries) in the destination leg of the trip script. To achieve this, the generator is given a mixed conceptual structure of the form:



where *X* is the actor of the answer key and *Y* is the script summary (or summaries) from the destination leg of the trip script. This general answer format is applied to the script instantiations in the destination leg of the script to produce a conceptual answer for the generator. Thus the answer 'Because John wanted to go to Leone's,' is produced in response to 'Why did John go to New York?'

If the question concept is matched in the destination leg of the script, the answer is produced by finding the MainCon of the script instantiation whose summary matched the question concept. The MainCon of a script instantiation is the conceptualization corresponding to the main act of that script. See Cullingford for a description of MainCons in scripts [Cullingford 1976, 1977]. When the MainCon is found, the conceptual answer is produced by generating:



where *Y* is the MainCon, and *X* is the actor of the MainCon. In the case of the restaurant script, the MainCon is the patron eating his meal. So if we were to ask 'Why did John go to Leone's?' the question concept (John went to Leone's) would match the script summary for the instantiation of the restaurant script (John went to Leone's). Since this match is made in the destination leg of the trip script, the question is answered by manipulating the MainCon of the restaurant script to get 'Because John wanted to eat some lasagna.'

If the matching search succeeds in the returning leg of the trip script, a canned answer is returned by generating a conceptualization which is translated: 'Because ACTOR(SS) wanted to go home,' where ACTOR(SS) is the actor from the matched script summary. So if the Leone's story described John leaving Leone's to take a subway and then a bus from New York, the questions 'Why did John leave Leone's?' and 'Why did John leave New York?' would both be answered 'Because John wanted to go home.'

5.1.2 Planning Structure Retrieval Heuristics

If no match is found for the question concept on the level of script structures, planning stuctures are tried next. The matching search examines the causal chain representation and plan-related inferences (if there are any) looking for an answer key. If the question concept is matched, plan-based retrieval heuristics are invoked in an effort to produce an answer. These heuristics are designed to produce a goal-oriented answer whenever possible (see section 2.2).

Once the answer key is found in the story representation, its immediate causal antecedents are extracted from the causal chain of events. If there are no immediate causal antecedents, then the memory search on the level of planning structures has failed. Otherwise each of these antecedents is checked to see if it is tagged as an act which instantiates some plan. Any act which is tagged as being part of a plan instantiation will have a pointer to the immediate goal motivating that plan. If a plan instantiation is found, its immediate goal forms the basis of the conceptual answer. In the dragon story where John rescues Mary from a dragon the Causal Antecedent questions, 'Why did John kill the dragon?' and 'Why did John get on his horse?' are both answered in terms of plan instantiated goals:

Q: Why did John kill the dragon?
A: Because he wanted Mary to not die.

Q: Why did John get on his horse?
A: Because he wanted to be near Mary.

If none of the direct antecedents yields a goal via plan instantiation, then no goal-oriented answer can be found. In this case, the same list of antecedents is searched for a concept which was motivated by a state. When one is found, its causal motivation is used to form the conceptual answer. This is how the question about Mary marrying John (in the dragon story) is answered:

Q: Why did Mary agree to marry John?
A: Because she was indebted to him.

The manipulation of conceptualizations to form a final conceptual answer is similar to those described previously for script structure retrieval heuristics. These plan-based retrieval heuristics will be discussed further in the section on retrieval heuristics in PAM (section 8.3.2).

5.1.3 Causal Chain Retrieval Heuristics

These heuristics are employed if the question concept has been found in the causal chain representation but no answer could be found using plan-based retrieval heuristics. In this case we expect an answer to be found on the basis of (1) inferences made by the script applier at the time of understanding, or (2) scriptal world knowledge.

5.1.3.1 Non-Standard Inference Search

When the script applier generates a causal chain representation, it tags some conceptualizations as being inferences. This tag is used to indicate a lesser degree of certainty about these conceptualizations. There is another tag for conceptualizations which were found in the main path of a script. For example, looking at a menu is in the main path of the restaurant script, but leaving because you have to wait for a table is not. For a description of different inference paths in scripts see [Cullingford 1977]. If the answer key has a causal antecedent which is tagged as an inference and is not tagged as a main path conceptualization, then this non-standard inference is used to generate a conceptual answer. For example, in the Leone's story, if we ask 'Why couldn't John pay the check?' the question concept (John couldn't pay the check) is matched against the conceptualization in the causal chain representing 'John discovered he couldn't pay the check.' This answer key has an inference as a causal antecedent, 'John had no money,' which is used to form a conceptual answer. The final answer is 'Because John had no money'. A mixed concept of the form 'Because CA' (where CA is the selected causal antecedent) is used to construct the final conceptual answer.

5.1.3.2 Interference Search

If no inference-tagged causal antecedent is found, we then check to see if the answer key is itself tagged as a script resolution [Lehnert 1977, Cullingford 1977]. If so, its corresponding interference concept is picked up to form an answer. 'Why did John have to wash dishes at Leone's?' is a question answered in this manner. John washing dishes is a script resolution for the interference of not being able to pay the check. So the question is answered 'Because John discovered he couldn't pay the check.'

5.1.3.3 Script-Internal Goal Structures

If all these other heuristics fail, we then resort to answering the question on the basis of scriptal world knowledge alone. This is done by analyzing the answer key in terms of its place within a specific script. All scripts contain a goal/subgoal structure which is strongly correlated with hierarchies of scenes within a script and sub-script maincons of each scene. For example, 'Why did John order lasagna?' asks about the maincon of the ordering scene in the restaurant script. The script-based answer 'Because he wanted to eat lasagna,' references the maincon of the eating scene. Of course a script-based answer of this sort is acceptable only when there is no other information around. Had the story mentioned that John always ate lasagna at Italian restaurants, this information would provide a better answer than the obvious connection of wanting to eat what you order.

While some retrieval heuristics on the level of script structures are very script-specific, the heuristics which look for goals are generalizable across scripts. While each script has a different goal structure, there are some structural features, like scenes and maincons, which are universal to all scripts. The rules which manipulate these structural features are very general. For example, if scene N does not come after the main act of the script, then the goal behind the maincon of scene N is the maincon of scene N+1.

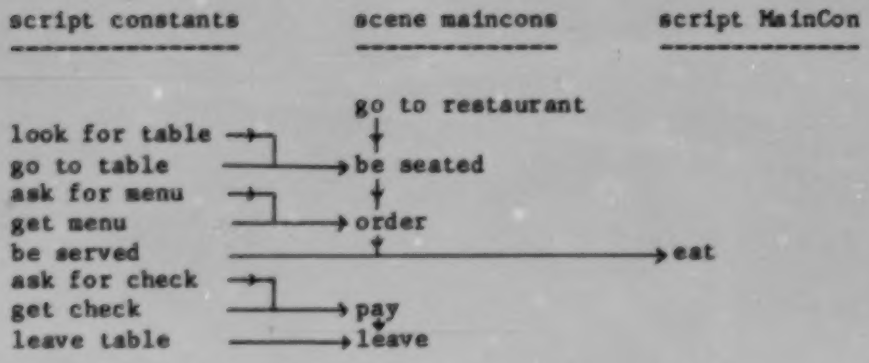


Figure 5

Restaurant Script Goal Hierarchy

If the question concept cannot be matched against any of the conceptualizations in the script structures, plan-related inferences, or causal chain, then no conceptual answer can be produced and the memory search returns a conceptualization to the generator which translates into 'I don't know.' The same answer is returned if the question concept is matched but all of the above heuristics fail to produce an answer.

5.2 Goal Orientation

The set of retrieval heuristics for Goal Orientation questions overlaps with those heuristics for Causal Antecedent questions to a large extent. This is quite reasonable since a Goal Orientation question is a particular type of Causal Antecedent question. This being the case, it is easiest to describe retrieval for Goal Orientation questions in terms of how it differs from Causal Antecedent retrieval.

Like Causal Antecedent questions, Goal Orientation questions can find answers in either the causal chain representation, planning structures, or script structures. As before, the first step is to match the question concept against a conceptualization in the story representation.

5.2.1 Script Structure Retrieval Heuristics

The matching search begins with script structures and uses the same script-specific heuristics that Causal Antecedent questions used for script structures. Thus far, all of the script-specific heuristics which have been developed for Causal Antecedent questions return answers which identify a goal orientation. As long as this continues to be the case, retrieval for Causal Antecedent and Goal Orientation questions will be identical on the level of script structures.

5.2.2 Planning Structure Retrieval Heuristics

If the matching search fails at the script structure level, heuristics applicable to planning structures are invoked. Here there is a slight departure from the plan-related retrieval processes used for Causal Antecedent questions. Once an answer key is found in the story representation, its immediate causal antecedents are extracted from the causal chain of events. If there are no immediate causal antecedents, then the memory search on the level of planning structures has failed. Otherwise each of these antecedents is checked to see if it is tagged as an act which instantiates some plan. Any act which is tagged as being part of a plan instantiation will have a pointer to the immediate goal motivating that plan. If a plan instantiation is found, its immediate goal forms the basis of the conceptual answer. A manipulation of the immediate goal identical to those used in Causal Antecedent questions becomes the final conceptual answer which is passed to the generator. In this way we can answer:

> Q: For what purpose did John kill the dragon?
> A: So Mary would not die.
>
> Q: For what purpose did John get on his horse?
> A: So he could be near Mary.

in the context of the dragon story:

> John loved Mary but she didn't want to marry him. One day, a dragon stole Mary from the castle. John got on top of his horse and killed the dragon. Mary agreed to marry him. They lived happily ever after.

If no plan instantiation was found for a Causal Antecedent question, a search was executed which looked for a motivating state. In Goal Orientation questions this search is not conducted. Since Goal Orientation questions ask about the goal or ultimate intentionality behind an action, they cannot be answered in terms of motivating states. It is reasonable to answer:

> Q: Why did Mary agree to marry John?
> A: Because she was indebted to him.

But it is not appropriate to answer:

> Q: For what purpose did Mary agree to marry John?
> A: Because she was indebted to him.

So if no plan instantiation is found, no further heuristics are invoked, and the matching search for a Goal Orientation answer has failed on the planning structure level.

5.2.3 Causal Chain Retrieval Heuristics

Again we have a slight departure from the causal chain retrieval heuristics used for Causal Antecedent questions. Neither the non-standard inference search or the interference search produce goal oriented answers, so neither of those searches are executed for Goal Orientation questions. It would not be appropriate to answer:

Q: For what purpose did John order a hamburger?
A: Because the waitress said they had no hot dogs.

But the script-internal goal structures do provide goal oriented answers. So the goal structure of a script is used to answer Goal Orientation questions:

Q: For what purpose did John order a hamburger?
A: So he could eat a hamburger.

If either no answer key is found, or if all of the above heuristics fail to find an answer, a conceptualization is given to the generator representing 'I don't know.'

5.3 Enablement

Enablement questions are a special case of Causal Antecedent questions. As such, their processing is similar to that for Causal Antecedent questions. Answers to Enablement questions may be found in either the script structures, planning structures, or on the causal chain level. The matching search examines the entire story representation at each level to find an answer key.

5.3.1 Script Stucture Retrieval Heuristics

Each script has specific entry conditions which must be satisfied before a default path script execution can be achieved. For example, the entry conditions for the restaurant script are that the patron be hungry and that he have some money. It is possible to execute the script without these states being satisfied, but if a script is entered with an entry condition violation, the script applier anticipates specific difficulties within the script which are not part of the default path script instantiation. For example, in the Leone's story, John goes into Leone's without any money. The inference about John not having money has been generated before John enters Leone's. So when the script applier hears that John can't pay the check, it has anticipated this particular interference and is ready to link this anomalous turn of events with the entry condition violation. For a more complete description of this process see [Cullingford 1977].

If the story had made no mention of any difficulty with the check, the script applier would have had to make an inference about an alternative entry condition which could account for John being able to pay the check. In this case, it would check the credibility assigned to the inference that John has no money, determine that there is some uncertainty about it, and conclude that John must have had some money with him after all. The inference that John had no money would then be eliminated from the story representation.

In one version of the Leone's story we are told that John took a bus back to New Haven after washing dishes at Leone's. In this case the inference about John not having any money has been substantiated by the fact that John couldn't pay the check at Leone's. But the default entry condition for the bus script is having some money. So when the bus script is activated, the script applier anticipates some difficulty since the script's entry condition is not satisfied. Yet 'John took a bus to New Haven,' without any further mention of the bus trip creates a default path instantiation of the bus script. The script applier must somehow reconcile this with the fact that the entry condition for that script was violated. It first checks the credibility assigned to the conceptualization representing John having no money. But this concept is now tagged with a credibility value of absolute certainty. So it can't delete this concept from the story representation. It then looks to see if there are any alternative entry conditions which can replace the condition of having money. When it looks for an alternative entry condition it finds a resolution for the apparent inconsistency: in order to execute a default path through the bus script, one could have a bus ticket instead of money. The script applier then concludes that this alternative entry condition must have been satisfied, and incorporates this inference in the story representation with less than absolute certainty.

When the matching search is conducted for an Enablement question on the level of script structures, the question concept is checked against script summaries of all instantiated scripts in the story representation. If an answer key is found in one of the script summaries, the entry conditions for the script are examined. If an alternative entry condition is found, and the credibility value assigned to it is less than absolute certainty, the final conceptual answer is produced by generating 'Probably EC,' where EC is the entry condition conceptualization. This is how the question about the return bus ride is answered in the Leone's story:

Q: How was John able to take the bus to New Haven?
A: Probably John had a ticket.

5.3.2 Planning Structure Retrieval Heuristics

The plan applier decomposes goal oriented behavior into plan instantiations for goals and subgoals. This organization allows us to determine when an action has been carried out in order to further a specific plan. If the matching search can locate the answer key in the story representation, it checks to see if the conceptualization matched has a SUBGOAL link pointing to a subordinate goal whose achievement contributes to or furthers the satisfaction of the answer key goal. If such a pointer exists, an answer is generated by concatenating subgoal conceptualizations together. If only one subgoal exists, the answer refers to the single subgoal. This heuristic would be used to answer:

Q: How was John able to prevent Mary from dying?
A: John killed the dragon.

5.3.3 Causal Chain Retrieval Heuristics

If the previous heuristics fail to produce an answer, the answer key is located in the causal chain representation and its direct causal antecedents are examined. If one of the causal antecedents is connected to the matched conceptualization by an ENABLES link, that antecedent is taken to be the conceptual answer. Enablement links are generated by the script and plan appliers when either (1) the chronology of the causal chain calls for an enabling relationship, or (2) a necessary enabling condition is recognized for a specific act. This heuristic would be used to answer a question like:

Q: How was John able to leave a tip?
A: John had some money.

If no conceptualization in the story representation is found to match the question concept, or if none of the heuristics described produce an answer, the generator is given a conceptualization corresponding to 'I don't know.'

5.4 Causal Consequent

Answers to Causal Consequent questions are found on the level of either script structures or causal chain representation. When a Causal Consequent question is answered on the script structure level, the answer is described in terms of a broad overview of the story. When it is answered on the causal chain level, a much finer degree of detail within the story is described. Whenever a question can be answered on varying levels of detail, it is appropriate to answer on the same level of detail in which the question was asked.

A script structure answer is sought first since a question which can be answered on that level is one which references an entire script instantiation and is therefore asking about something in terms of an overview. If a question concept fails to be found on the script structure level, it is asking about something within a particular script and therefore deserves to be answered in terms of acts within the script instantiation, acts which are on the same finer level of detail as the question itself.

An important feature of Causal Consequent questions is that they are ambiguous in terms of chronological ordering. In the context of the Leone's story,

Q: What happened when John got off the bus?

could be reasonably answered with either:

A: He thanked the driver, or
A: He went to the subway.

The first answer describes an event which happened before John actually got off the bus, and the second answer describes an event which took place after John got off the bus. In addition to answers which strictly precede or follow the question concept, an answer to a Causal Consequent question can occur at the same time as the act in question:

> Q: What happened when John took the subway?
> A: A thief picked John's pocket.

Causal Consequent questions are actually questions which say, 'tell me about something that happened at about the same time as (blitch) took place.' Of course the best answers are those which describe the most interesting thing possible, and the retrieval heuristics must incorporate some notion of relative interest values when searching for an answer.

5.4.1 Script Stucture Retrieval Heuristics

The matching search begins with script structures, trying to find the answer key in the script summaries of those scripts which were instantiated for the story representation. If the question concept matches one of these script summaries, the question is asking about a relatively rough time reference: acceptable answers can describe events which took place anywhere within the course of the referenced script instantiation. 'What happened when John took the subway?' is one such question. The question concept (John took the subway) matches the script summary for the subway trip (John went to New York by subway) and is therefore asking for events which occurred anytime during the subway trip.

5.4.1.1 Weird Event Search

When the answer key is a script summary, the memory search first looks to see if the corresponding script instantiation contained any events which were recognized as weird events by the script applier. A weird event is a conceptualization which the script applier cannot find in any of its currently active scripts, and which itself triggers a script which is unusual or inappropriate in the context of those scripts which are currently active. See [Cullingford 1977] for a description of weird events within the context of a script. In the Leone's story, John being pick-pocketed is recognized as a weird event and is tagged as such in the subway script instantiation. If a weird event is found, it is picked up as the conceptual answer to the question. This is how the Causal Consequent question about the subway trip is answered:

> Q: What happened when John took the subway?
> A: A thief picked John's pocket.

5.4.1.2 Interference/Resolution Search

If no weird event is found within the script in question, the memory search looks to see if anything mildly interesting occurred during that script. It does this by checking for interference/resolution pairs. If there were one or more interference/resolution pairs, a conceptual answer is formed by concatenating them together in the form 'I1 and so R1, and then, I2 and so R2, and then . . .' where In and Rn are the nth interference and resolution in that script. This is how we answer:

> Q: What happened when John went to Leone's?
> A: John discovered that he couldn't pay the check and so he had to wash dishes.

5.4.1.3 Main Act Search

If the script has no weird events or interference/resolution pairs, then a test is executed to see if the script summary matches the main act of the script. In some scripts the main act and the script summary convey the same information (e.g. the pickpocket script) but in others they are different. If it turns out that the main act is different from the script summary, then the main act of the script is returned as the conceptual answer.

> John went to Leone's. He ordered lasagna. When he left he gave the waitress a large tip.
>
> Q: What happened when John went to Leone's?
> A: John ate lasagna.

5.4.2 Causal Chain Retrieval Heuristics

If the script structure memory search fails to produce an answer key, then a matching search begins on the causal chain representation. If the question concept cannot be found in the causal chain, the generator is given a conceptualization representing 'I don't remember anything.'

5.4.2.1 Default Path Departures

If a match is made, the memory search checks to see if the answer key is an instantiation of a maincon in a script scene. Most scripts are partitioned into scenes and each scene has one main act which is of primary importance. For example, the restaurant script has scenes for entering, seating, ordering, eating, paying, and leaving. See [Cullingford 1977] for a detailed description of script scenes. If the answer key does describe the main act of a scene, then the memory search examines each act of the script instantiation which is a part of that scene, looking for a conceptualization which is not tagged as an act on the default path of the script. If it finds an act which did not come from the default path, it returns that conceptualization as the conceptual answer.

For example, the main acts of the bus script scenes are getting on, sitting down, and getting off. So if we ask 'What happened when John got off the bus?' in the context of the Leone's story, the underlying question concept (John got off the bus) matches a concept in the causal chain which is tagged as the main act of the getting-off scene. Other conceptualizations from the getting-off scene are then checked to see if there was one which did not come from the default path of the bus script. One such conceptualization is found and it becomes the answer:

Q: What happened when John got off the bus?
A: John thanked the driver.

The same default path departure processing answers the question:

Q: What happened when John sat down on the bus?
A: John talked to an old lady.

5.4.2.2 Resolution Search

The memory search next checks to see if the answer key was tagged by the script applier as a script interference. If so, its corresponding resolution is returned as the conceptual answer:

Q: What happened when John couldn't pay the check?
A: The management told John he would have to wash dishes.

5.4.2.3 Chronological Consequent Search

If no answer is produced by the other heuristics, but an answer key was found in the causal chain, then the next act in the causal chain is taken as the conceptual answer:

Q: What happened when John ordered lasagna?
A: The waitress took the order to the chef.

Q: What happened when John told the waiter he couldn't pay the check?
A: The management told John he would have to wash dishes.

In an early version of SAM, Causal Consequent questions were answered by picking up all the conceptualizations following the question concept in the causal chain up to the next conceptualization which was explicitly stated in the story:

Q: What happened when the hostess gave John a menu?
A: John read the menu, the waiter saw that John was at the table, the waiter went to the table.

Q: What happened when John ordered the hamburger?
A: The waitress gave the order to the cook, the cook prepared the hamburger, the cook gave the hamburger to the waitress, the waitress served John the hamburger.

This heuristic was designed primarily as a way of showing off all of the inferences SAM makes. It was never intended to be taken as a serious model of what people do or as a heuristic for providing natural answers. One of its immediate faults lies within its reliance on knowing whether or not a conceptualization in the story representation was explicitly mentioned in the text of the story. A number of psychology experiments [Bower 1976, Bransford & Franks 1971] have shown that people cannot differentiate what they are explicitly told from what they infer. Furthermore, the type of information which gets confused appears to be exactly the mundane scriptal information with which we are working. Since our question answering model is attempting to model human cognitive processes, we do not want to develop memory mechanisms which rely on capabilities people do not have. Consequently, this early heuristic has been replaced by the chronological consequent search which returns the single concept immediately following the answer key in the causal chain.

5.5 Verification

The initial memory search for a Verification question is quite straightforward. A matching search is executed which examines all conceptualizations in the script structures, planning structures, and causal chain representation. If a match is made, the credibility value of the matched conceptualization is checked. If it has absolute certainty, the answer 'Yes' is passed to the generator. If it has a credibility which is less than absolutely certain, the answer 'Probably' is returned. If no answer key is found the initial answer is 'No.'

Verification questions often carry Elaboration Options from Content Specification (see Chapter Four). After an initial answer of yes, no or probably is produced, the memory search checks to see if the question is carrying an Elaboration Option or Category Trace Instruction. If not, the initial answer is the final conceptual answer. Otherwise an elaboration must be produced according to the instructions in the Elaboration Option or Category Trace Instruction.

If a Verification question is carrying an Elaboration Option, it will specify two things for the memory search: the Answer Criteria (AC), and the Elaboration Instructions (EI). The initial response is compared to the Answer Criteria. If the Answer Criteria test is passed, the Elaboration Instructions are executed. If the Answer Criteria test fails, the initial answer becomes the final conceptual answer which is passed to the generator. At the present time three Elaboration Options applying to Verification questions have been implemented: the Verification Option, the Short Answer Option, and the Correction/Explanation Option [see section 4.2]. These options are responsible for answers like:

Q: Did the waitress give John a menu?
A: No, the hostess gave John a menu.
(Correction/Explanation Option)

Q: Did John pay the check?
A: No, John was angry because the hamburger was
burnt and so he left.
(Correction/Explanation Option)

Q: Did John order lasagna?
A: Yes, John ordered lasagna.
(Verification Option)

Q: Did John go to New York?
A: Yes, John went to New York by bus.
(Verification Option)

Q: Did John go to the Friar's Club?
A: No.
(Short Answer Option)

5.6 Disjunctive

Disjunctive questions are similar to Verification questions in terms of the retrieval heuristics used to answer them. It is useful to think of Disjunctive questions as being Verification questions, but with multiple question concepts instead of one.

Initially, a matching search of the entire story representation (script structures, planning structures, and causal chain representation) is conducted for each of the question concepts until one of them is matched. As soon as a match is made, the matched concept is passed to the generator as the final conceptual answer. If no match is made, the initial answer is 'Neither.' Once an initial answer is produced, the question is checked to see if there is an attending Transform Trace Instruction or Elaboration Option.

There is a focus problem with Disjunctive questions. Occassionally a Disjunctive question can be answered simply Yes or No depending on whether or not any of the conceptualizations in the question concept can be verified.

Q: Did John or Mary go shopping?

may be asking:

Q: Who went shopping - John or Mary?

or it might be asking:

Q: Did anyone go shopping?

in a context where John and Mary are the only obvious candidates and the questioner is more interested in the act of shopping than the actor performing the act. Knowing which way a Disjunctive question should be answered depends on contextual factors and inferences about what the questioner is really interested in.

It is the case, however, that an answer which is derived by passing the answer key to the generator will always be adequate - at its worst, it will supply more information than the questioner was looking for. Saying that Mary went shopping will satisfy the questioner regardless of whether he was interested in who went or just that someone went. The focus issue for Disjunctive questions is therefore less crucial than it is in other conceptual question categories. (These problems will be pursued in Chapter Six).

5.7 Instrumental/Procedural

Answers to Instrumental/Procedural (I/P) questions may be found in either the script structures, planning structures, or on the causal chain level. The matching search begins with script structures. Not all scripts have script-specific retrieval heuristics for I/P questions. The only script implemented thus far which does have special heuristics for I/P questions is the trip script.

5.7.1 Script Structure Retrieval Heuristics

The matching search over script structures examines script summaries for each script which appears as a top level instantiation in the going or returning legs of the trip script. If the question concept matches one of these summaries, an answer is produced by combining other script summaries from the trip script structure. If the answer key is found in the going leg of the trip, the answer is formed by concatenating all of the summaries for each script in the going leg of the trip up to and including the matched summary:

Q: How did John get to New York?
A: John went to New York by bus.

Q: How did John get to Leone's?
A: John went to New York by bus and then he went
to Leone's by subway.

If the question concept matches a script summary from the returning leg of the trip, the same heuristic is used, but with the script summaries in the returning leg of the trip:

Q: How did John leave Leone's?
A: John went from Leone's by subway.

Q: How did John get to New Haven?
A: John went from Leone's by subway and then John
went from New York to New Haven by bus.

5.7.2 Planning Structure Retrieval Heuristics

If no answer is found in the script structures, inferences from planning structures are invoked. On this level, retrieval for I/P questions is identical to retrieval for Enablement questions. If the matching search can locate the answer key in the story representation, it checks to see if the answer key has a PLANACT link pointing to a subordinate goal. If such a pointer is found, an answer is produced by concatenating the PLANACT conceptualizations:

Q: How did John prevent Mary from getting hurt?
A: John killed the dragon.

Retrieval from planning structures for Enablement and I/P questions is identical since there does not seem to be a conceptual distinction between those plans which are enablements to a parent goal and those plans which are instrumental to a parent goal. At the present time there is only one relationship between a goal and its plans: a plan contributes to the achievement of its parent goal. In the questions and stories considered, Enablement and I/P questions appear to be conceptually equivalent when the answers are derived from goal/plan relationships. The questions:

Q: How was John able to prevent Mary from getting hurt?
Q: How did John prevent Mary from getting hurt?

can both be answered:

A: He killed the dragon.

If at some point a question is encountered which demands a distinction between instrumental plans and enabling plans, the representation will be forced to reflect this difference and the retrieval heuristics will be altered accordingly.

5.7.3 Causal Chain Retrieval Heuristics

On the causal chain level, the matching search runs through the causal chain looking for an answer key. If one is found, the answer key is checked to see if it has an Instrument slot. If so, that conceptualization is returned as the final conceptual answer:

Q: How did John get to the table?
A: John walked to the table.

If no Instrument slot is found in the matched conceptualization, or if no conceptualization in the causal chain is matched, a conceptualization is passed to the generator representing 'I don't know.'

5.8 Concept Completion

Concept Completion questions require little more than a matching search to be answered. The question concept for a Concept Completion question has an unknown conceptual component. In the memory search, this unknown component is treated as a wild card; it will match anything. The matching search examines all levels of the story representation. Script structures, planning structures, and the causal chain representation are all searched for an answer key.

When a match is made, short answers are produced if the Single Word Option is in effect (see 4.2).

Q: Who gave John a menu?
A: The hostess.

Otherwise, the entire answer key is generated to produce a long answer:

Q: Who gave John a menu?
A: The hostess gave John a menu.

5.9 Expectational

Expectational questions are unlike all of the conceptual question categories discussed so far in terms of the initial memory search processing which must take place. For all of the other question categories a matching search was initially executed to locate the question concept in the story representation. But an Expectational question asks about something which didn't happen. And a conceptual story representation contains only those things which did take place. So an answer key for an Expectational question cannot be found in the story representation. Therefore Expectational questions do not initially conduct a matching search for the question concept like other question categories. A matching search is conducted, but it operates on a data structure which is created at the time of the memory search to augment the story representation.

A detailed account of Expectational questions and the memory search processes which answer them is given in Chapter Seven.

5.10 Judgemental

Judgemental questions are similar to Expectational questions insofar as they must access information outside of the story representation in order to produce an answer. Questions like 'What should John do now?' require the answerer to project himself into John's place (i.e. into the situational context of the story) and make a projection concerning John's behavior on the basis of whatever scripts and plans the answerer has at his disposal.

These questions have been implemented by Jaime Carbonell in his script-based version of the Goldwater machine [Carbonell 1977]. The POLITICS program currently processes Conceptual Dependency input and produces Conceptual Dependency answers. Given an input statement and some questions, POLITICS responds with answers which reflect the political ideology of a right-winger:

INPUT: Russia massed troops on the Czech border.

Q: What will Russia do next?
A: RUSSIA MAY ORDER ITS TROOPS INTO CZECHOSLOVAKIA.

> Q: What can the United States do?
> A: THE UNITED STATES CAN DO NOTHING, IT CAN INTERVENE MILITARILY IN CZECHOSLOVAKIA BY SENDING TROOPS, OR IT CAN INTERVENE DIPLOMATICALLY BY TALKING TO RUSSIA ABOUT CZECHOSLOVAKIA.
>
> Q: What should the United States do?
> A: THE UNITED STATES SHOULD INTERVENE MILITARILY.

To implement Judgemental questions for SAM and PAM, we would have to set up an interactive communication between the question answerer and the script applier or plan applier (much like the interaction implemented for Expectational questions which will be described in Chapter Seven). In this interaction the script or plan applier would be given a processing state encountered at some point during understanding, along with possible goals specified in the question, and a request to project the predicted behavior of whatever character was in question.

For example, suppose we have read the following story:

> John went into a restaurant. The waitress gave him a menu and he decided he wanted a hamburger.

Now suppose we are asked:

> Q: What should John do now?

To answer this question we would ask the script applier for the next conceptualization involving John:

> A: John should order a hamburger.

Suppose we had read:

> John loved Mary but she didn't want to marry him. One day a dragon stole Mary from the castle. She was never seen again.

and were asked:

> Q: What should John have done to save Mary?

To answer this question the plan applier would have to be given (1) the story representation as it stood before hearing that Mary was never seen again, and (2) a goal on the part of John to save Mary. It could then predict a plan which John could invoke for attaining his goal. This plan would be used to produce an answer:

> A: John should have killed the dragon.

No difficulties are anticipated in implementing these questions. Many of the mechanisms which would be needed already exist for the Expectational questions. These questions have not been implemented for SAM or PAM largely because they have been implemented elsewhere.

5.11 Quantification

Quantification questions are very much like Concept Completion questions but their answers are found in memory tokens rather than in the chronological story representation. A fact like how many people were in Leone's would be stored under the memory token for Leone's patrons. It would not be placed in the script structures, planning structures, or causal chain conceptualizations.

The Internalization processing which follows the initial parse should be able to identify which memory token is being referenced, so the only work remaining for the retrieval heuristics is to look up the appropriate property under that memory token. For example, if the story had mentioned that there were twenty people on the bus going to New York, then a memory token for that group of people would be created at the time of understanding. Say the memory token was given the pointer GN005. Under that memory token would be the property NUMBER, and the value assigned to the property NUMBER would be 20. Now if a question is parsed which asks 'How many people were on the bus going downtown?' the internalization program would be responsible for recognizing that this question references GN005. The internalized conceptualization representing the question would therefore be:

$$\text{GN005} \Longleftrightarrow \text{IS NUMBER VAL (*?*)}$$

The question concept consists of two parts, the referent and the unknown property:

(REF GN005 PROP NUMBER)

The retrieval heuristic is then very simple. The conceptual answer is produced by performing a GET (the function which accesses property list values) on the name GN005 and the property NUMBER.

5.12 Feature Specification

Feature Specification questions are answered the same way as Quantification questions. The question concept consists of a tokenized referent and an unknown property value. A simple retrieval on the property list of the specified memory token produces the answer.

Q: How old was John?

is parsed and internalized as:

$$\text{GN001} \Longleftrightarrow \text{IS AGE VAL (*?*)}$$

The interpreted question concept is (REF GN001 PROP AGE) and the answer is produced by getting the value of the property AGE under the name GN001. In the event that a fuzzy property reference is given in the question (What kind of dog is Rover?) the retrieval looks for any descriptive property which is not in a list of standard descriptors to avoid (SEX, AGE, COLOR, WEIGHT, NATIONALIY, etc.) The first property it finds which is not on this list is returned as the conceptual answer. This heuristic should be replaced by a type-specific default heuristic. For example, a memory token of type DOG should default to BREED when a general descriptor is needed. But such a type-specific search would require a taxonomy of memory token types. While such a taxonomy must eventually be proposed, it is too early to develop anything other than an ad hoc system of memory token hierarchies. The primitive heuristic described above will stand until a memory token taxonomy can be proposed.

5.13 Concluding Remarks on Retrieval Heuristics

The one conceptual question category which does not dictate retrieval heuristics are Requests. In a complete model of question answering, a large portion of the memory search should be devoted to the processing required to execute a Request. This aspect of the theory would naturally be limited to whatever simulated world the computer can function in. For example, Winograd's SHRDLU explored the Request aspect of memory searches in its manipulation of a blocks world [Winograd 1972].

Since the programs implemented in conjunction with QUALM are programs which function to demonstrate story understanding, there has been no natural opportunity for exploring Requests in terms of memory searches. The questions which people can ask about stories are Inquiries. These have formed the basis for the research presented here. In the context of story understanding, we need not concern ourselves with whether or not the computer is willing to light our cigarettes.

CHAPTER 6

FOCUS ESTABLISHMENT

The focus of a question is that conceptual component of the question to which attention is directed. When focus is misplaced, answers to questions appear to miss the point:

Q1: Why does Carter carry his own luggage?
A1: He wants us to know he's one of the people.
A2: It would be silly for him to carry Billy's things around.

A1 answers a question which focused on the fact that the President is violating a social convention of his office: when you're president you don't carry your own luggage. The focus falls on Carter:

Q2: Why does Carter (of all people) carry his own luggage?

A2 answers the same question but with focus on the fact that people normally carry their own luggage instead of other people's. The focus here falls on whose luggage Carter carries:

Q3: Why does Carter carry his own luggage (instead of someone else's luggage)?

6.0 Introduction

If A1 seems to be a more natural answer to Q1 than A2, it is only because we know that Carter is President, and that presidents are not subject to the same social conventions as the rest of us. It would not make much sense to answer:

Q4: Why does John Doe carry his own luggage?
A4: He wants us to think he's one of the people.

unless there is something special about John Doe which suggests that he is not just one of the people. This particular example of focus establishment is based on general knowledge about the world and will be discussed in section 6.1.4. But first, we will describe other ways that focus manifests itself.

6.1 Different Kinds of Focus

Attention can be directed to a particular component of a question by many different means. Four general strategies used by speakers for focus placement are:

1) Stress Intonation Patterns
2) Syntactic Construction
3) Contextual Determinants
4) Knowledge-Based Determinants

The first strategy, stress intonation, is a function of acoustic processing and cannot be detected in written language without the use of italics, underscoring, or some other visual device designed to mimick spoken stress. The second strategy, syntactic construction, is a grammatical device. The last two strategies, contextual focus and knowledge-based focus, are functions of conceptual processing which cannot be discussed without reference to memory processes and information in memory.

6.1.1 Stress Intonation Patterns

One way that a question can be constructed with a clear focal assignment is by using various intonation patterns. Of course this does not occur in written language but it is used all the time in spoken language.

Q2: Did the WAITRESS bring John a menu?

A stress pattern for Q2 which emphasizes the word 'waitress' by a rise in tonal frequency and perhaps volume will serve to place the focus of the question on the waitress. With the proper intonation, Q2 is functionally equivalent to asking:

Q3: Was it the waitress who brought John a menu?

in terms of where the attention is forced and what kinds of answers will be appropriate.

In actual speech, intonation can be combined with syntactic construction to express attitudes of irony, disbelief, or sarcasm. To see how intonation can turn around the meaning of entire sentences, consider two readings of a fictional telegram:

Joseph Stalin
Kremlin
Moscow

You were right and I was wrong. You are the true

heir of Lenin. I should apologize.

Trotsky

The exact same telegram read with feeling becomes an entirely different message:

Joseph Stalin
Kremlin
Moscow

YOU were right and I was WRONG? YOU are the true heir of Lenin? I should *apologize*??!!

Trotsky!!!!!!!!

6.1.2 Syntactic Constructions

There are syntactic constructions which function as focus operators:

Q3: Was it the waitress who brought John a menu?

This question is constructed to place emphasis on the conceptual component of the waitress. Any appropriate answer to Q3 will address this aspect of the question concept. Q3 can be answered naturally with:

A3a: No, the hostess brought John a menu.

but it would be odd to answer Q3 with:

A3b: No, the waitress brought John a hamburger.

Q3 effectively asks 'Did the waitress bring John a menu? If not, who brought John a menu?' The question tends to convey a presupposition that somebody brought John a menu, the question is who? So the focus for Q3 falls on the actor who brought John a menu.

6.1.3 Context and Focus

Syntactic constructions and stress intonation patterns are aspects of a question that the parser must recognize. While syntactic construction and stress intonation patterns can be used to establish the focus of a question, there are also times when the focus does not rely on anything which the parser can be expected to handle. It is often the case that the context of a question is essential in establishing the focus of a question. To see how context can affect focus, consider a question asked in the contexts of two different stories:

CONTEXT 1

> John had just bought a new car. He was so happy with it that he drove it at every possible opportunity. So last night when he decided to go out for dinner, he drove over to Leone's. When he got there he had to wait for a table . . .
>
> Q4: Why did John drive to Leone's?

Appropriate answers to this question are 'Because he just got a new car and he liked to drive it whenever he could,' or 'Because he was very happy with his new car,' or Because he enjoyed driving,' etc, etc. The point is that the question here is interpreted to be asking about driving. The focus of the question is on the transportational instrument used when John went to Leone's. Now consider another story:

CONTEXT 2

> John had a crush on Mary. But he was so shy that he was happy to just be in her proximity. So he was in the habit of following her around a lot. He knew that she ate at Leone's very often. So last night when he decided to go out for dinner, he drove over to Leone's. When he got there he had to wait for a table . . .
>
> Q5: Why did John drive to Leone's?

The question now has a different meaning. Appropriate answers are 'Because he knew that Mary ate there,' or 'Because he hoped to run into Mary there,' or 'Because he wanted to see Mary,' etc, etc. Here the question has been interpreted to be asking about Leone's. The focus of the question is on John's destination.

Since Q4 and Q5 elicit different answers, they cannot be conceptually equivalent questions. Yet lexically, Q4 and Q5 are identical questions. They differ only in terms of interpretive focus. This assignment of focus must be a function of the context in which the question occurs. The conceptual representation of a question is not complete if it does not include focus specification whenever appropriate. It follows that the conceptual represention of some questions must therefore depend on the context in which the question occurs.

When the focus of a question is sensitive to context, questions tend to be interpreted in terms of what information is present in memory. In the first context we don't know why John chose to go to Leone's in particular but we do know why he drove. In the second context we have no information concerning John's choice of transportation but we do know why he elected to go to Leone's. People do not consider an alternative interpretation of a question which they cannot answer when there is a natural interpretation which can be answered. This suggests that there are questions where memory must be accessed before a full interpretation of the question is achieved. A discussion of this problem and some proposed solutions will be presented in Chapter Eight.

6.1.4 World Knowledge and Focus

In some cases the focus of a question can be established only by accessing world knowledge in memory and applying various inference processes in order to see where the question was likely to be focused. In these cases the establishment of focus occurs in the interpreter or the memory search.

For example, suppose I tell you that our mutual friend John roller skated to McDonald's last night. You may very well ask:

> Q6: Why did he roller skate to McDonald's?

and I could answer back:

> A6a: Because he was hungry.
> A6b: Because his bicycle was broken.

A6a addresses the question as an inquiry about John's destination. A6b answers the question in terms of John's mode of transportation. If A6a seems to be a funny answer it is because it addresses an unnatural focus assignment. Q6 is more naturally interpreted to be asking about John's roller skating rather than John's destination. This focus preference is a function of evaluating what is most interesting about the question. John going to McDonald's is far more commonplace than his roller skating (given the assumption that John is an adult). This interest evaluation must be done in terms of world knowledge and knowing what things are relatively common or unusual.

Of course if the reader knows that John is an eccentric who roller skates everywhere and never goes into McDonald's because he abhors fast food, then this knowledge will be used to understand that John going to McDonald's is more interesting than John roller skating.

Focus Rule #1:
SPECIFIC KNOWLEDGE HAS PRIORITY
OVER GENERAL KNOWLEDGE DURING
FOCUS ESTABLISHMENT

But in the absence of specific knowledge, there is a default hierarchy of world knowledge in terms of relative interest values. No matter where the interest evaluations come from, the same interpretive rule always holds: when different components of a question are competing for the focus it is natural to emphasize the most unusual aspect of the conceptual question.

Another knowledge based focus assignment is concerned with placing focus on the most variable aspect of a question:

Q7: Did the waitress bring John a menu?

Suppose the answer to this question is No and we want to elaborate the negative response.

A7a: No, the waitress brought John a hamburger.
A7b: No, the hostess brought John a menu.

A7a results when the focus of Q7 is placed on the menu. The elaboration is predicated on the belief that what is important about this question is what the waitress brought John. A7b results when the focus is placed on the actor, the waitress. Here the elaboration is produced by understanding Q7 to be concerned with who brought John the menu. When focus is assigned to the menu, Q7 carries a weak presupposition that the waitress brought John something. When focus is assigned to the waitress, the presuppossition that somebody brought John a menu. Focus is being placed on that component of the question concept which is most open to correction or variation.

Focus Rule #1:
FOCUS FAVORS
VARIATION OVER EXPECTATION

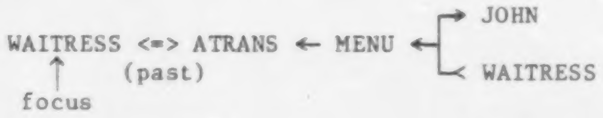
It may very well be the case that both A6a and A6b are perfectly correct answers. But one of them is liable to be more appropriate. Finding a focus which will result in the most approriate answer requires knowledge about the world, stereotypic occurrences, and points of variation within a stereotypic situation. We will return to this particular type of focus problem and propose a script-based processing solution for it in section 6.3.

6.2 When Focus is Established

Syntactic constructions and intonational patterns allow us to identify the focus of a question without any inferencing or higher memory processing. In either case, the parser can recognize the focus of the question and mark the component receiving emphasis in its resulting conceptual representation. In our computer models intonational patterns are not considered since we are processing written rather than spoken input. But syntactic constructions should be recognized by the parser when they function in terms of focus establishment. The parser should be able to input

Q1: Was it the waitress who gave John a menu?

and output a conceptualization which marks the focus of the question:



The current parser used in SAM and PAM recognizes and assigns focus in some cases. The parser design has not incorporated comprehensive focus recognition largely because of design priorities. For example, the recognition of various cleft sentence constructions is both a straightforward and relatively peripheral problem which is not expected to pose any interesting theoretical problems.

One of the basic processing principles which applies to the problem of focus establishment concerns the desirability of not doing something you don't have to do. Some questions can be answered without ever considering the focus of the question. If the answer to

Q6: Did the waitress bring John a menu?

is Yes, then it is not necessary to know the focus of the question. A memory search can be conducted which finds a concept corresponding to the waitress bringing John a menu, the answer Yes is returned, and no more processing is required. Focus is only needed to answer Q6 when the initial answer is No and an elaboration is desired. It is therefore appropriate to relegate focus establishment to the answer elaborator in cases like Q6 where focus is not needed unless an elaboration option is being exercised.

6.3 A Script-Based Focus Heuristic

The focus heuristic about to be described is executed when the Correction/Explanation Option is exercised to augment a negative response to a Verification question. The focus of the question must be identified in this case in order to correct the question concept. Before launching into a description of the actual heuristic, we will discuss various Correction/Explanation elaborations to see what is involved in producing these answers.

Suppose we are asking questions in the context of the following story:

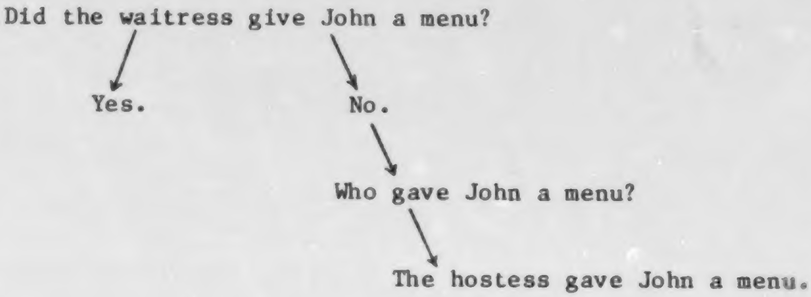
> John went to a restaurant and the hostess gave him a menu. When he ordered a hot dog the waitress said they didn't have any. So John ordered a hamburger instead. But when the hamburger came, it was so burnt that John left.

Q1: Did the waitress give John a menu?
A1: No, the hostess gave John a menu.

Q2: Did the waitress serve John a hot dog?
A2: No, the waitress served John a hamburger.

Q3: Did John eat the hamburger?
A3: No, the hamburger was burnt.

Each of these Verification questions has been answered with an appropriate elaboration. The problem we are concerned with is where these elaborations come from. If we consider Q1, Q2, and Q3, it becomes clear that each of the elaborations offered here are themselves answers to questions:



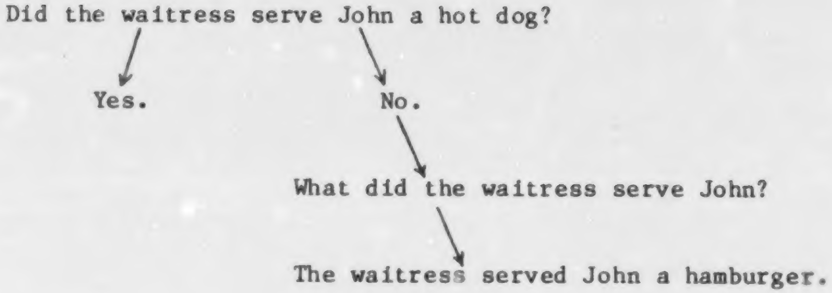
Did the waitress give John a menu?

Yes.

No.

Who gave John a menu?

The hostess gave John a menu.



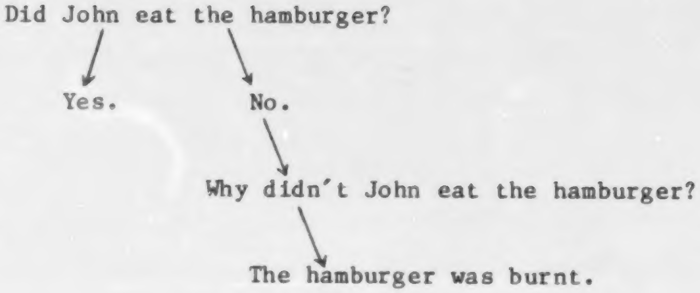
Did the waitress serve John a hot dog?

Yes.

No.

What did the waitress serve John?

The waitress served John a hamburger.



Did John eat the hamburger?

Yes.

No.

Why didn't John eat the hamburger?

The hamburger was burnt.

It appears that these elaborations are obtained by asking and answering some new question. So the problem of finding an elaboration becomes the problem of finding a question to ask (and answer). Once we have asked the right question, finding an answer is not hard: the secondary question can just be fed back into QUALM to be processed as if it were just another top level question. The difficulty is in asking the right secondary question. How do we know which question will lead to a good elaboration?

It should be clear that there are many secondary questions which could be generated. In fact, there are many secondary questions which would lead to correct elaborations.

Q4: Did the waitress give John a menu?
A4: No, the waitress gave John a hamburger.

A4 comes from answering the secondary question 'What did the waitress give John?' It is important to realize that there is nothing incorrect about A4 as a response to Q4. It is absolutely true that the waitress gave John a hamburger. Yet somehow this answer seems less natural or less appropriate than the answer, 'No, the hostess gave John a menu.' In generating a secondary question we must be concerned with finding the one which leads to the most natural response.

All secondary questions which lead to reasonable elaborations can be derived from the original question concept in some manner. A1 and A2 are the result of Concept Completion questions created from the original question concept. 'Who gave John a menu?' comes from replacing the waitress with an unknown Actor in the question concept underlying 'Did the waitress give John a menu?' 'What did the waitress serve John?' is the result of an unknown Object in place of the hamburger in the concept underlying 'Did the waitress serve John a hot dog?' A3 is a response to an Expectational question: 'Did John eat the hamburger? (No,) - Why not? (the hamburger was burnt.)' How do we know when to generate a Concept Completion question and when to generate an Expectational question? And if a Concept Completion question is to be used, which one?

For the moment, let's just worry about finding the right Concept Completion question. Each possible Concept Completion question is derived by replacing a conceptual component in the original question concept with an unknown. The problem we face is one of knowing which conceptual component should be replaced. Consider all the possible Concept Completion questions which could have been generated from Q1:

Did the waitress give John a menu?
(question concept)

```
             *?*                      ┌→ JOHN
WAITRESS <=> ATRANS ← MENU  ←─┤
                                      └< WAITRESS
```

Who gave John a menu?
(replace Actor slot)

```
                            ┌→ JOHN
*?* <=> ATRANS ← MENU ←─┤
                            └< *?*
```

What did the waitress do?
(replace Conceptual Act slot)

```
WAITRESS <=> *?*
```

What did the waitress give John?
(replace Object slot)

```
                                  ┌→ JOHN
WAITRESS <=> ATRANS ← *?*  ←──────┤
                                  └< WAITRESS
```

Who did the waitress give a menu to?
(replace Recipient slot)

```
                                  ┌→ *?*
WAITRESS <=> ATRANS ← MENU ←──────┤
                                  └< WAITRESS
```

By focusing on different conceptual components in the question, we can generate different secondary questions for elaborations. Some of these secondary questions will lead to an elaboration and some won't. For example, in the context of our original story 'What did the waitress give John?' can be answered and will therefore provide an elaboration. But 'Who did the waitress give a menu to?' cannot be answered (unless the answer is no one) and therefore does not lead to a good elaboration.

While many secondary questions can be generated, and more than one of these may lead to a correct elaboration, there is generally one question which leads to the most natural elaboration. Knowing which question will give the best elaboration is equivalent to knowing which conceptual component of the question should receive attention. There is an implicit focus in these questions which singles out the proper conceptual component. In Q1 the focus is on the waitress. In Q2 focus falls on the hot dog. The problem is how to identify the implicit focus in these questions.

There is a basic principle in question answering which can be used to guide all problems in focus establishment.

Focus Rule #3:
Focus should always fall on that conceptual component which is most interesting.

The only problem is to determine which component in a conceptual question is relatively interesting. To do this we must use knowledge about the world. We must use knowledge in order to find out what expectations the person asking the question has.

Once we've answered the question 'Did the waitress give John a menu?' with 'No,' we have contradicted an implicit expectation on the part of the questioner. There was some expectation that the waitress might have or should have given John a menu. In elaborating our answer, we wish to address this expectation and explain why it was violated. If we consider the source of this expectation, we can determine different degrees of certainty in the conceptual components of the expectation. These variations in certainty derive from the notion of script constants and script variables.

In every script there are a set of very strong expectations. When we hear that John went to a restaurant, we expect certain activities to have taken place. For example, we expect that John sat down, he got a menu, he ordered, he ate, and he must have paid the check. These acts are called script constants. If our expectations regarding a script constant are violated, we want to be able to account for the contradiction. So if we hear that John went to a restaurant but did not pay the check, we tend to want to know why not. Some explanation is expected and will be sought.

Within each of the expected script constants, there is often room for a certain amount of variation. We know that John must have gotten a menu, but it is not clear where the menu comes from. He might get it from the waitress, or the hostess, or it may be sitting on the table and he picks it up himself. The source of the menu is a script variable. Naturally what John orders and eats are script variables, as well as who brings him the check (it could be the waiter/waitress or it might be the host/hostess). Some script variables take default assignments in the absence of explicit information. For example, I would assume that the waiter/waitress brings the check unless I am told otherwise. But I would make no assumptions about what is eaten in the absence of any explicit information.

SCRIPT CONSTANTS

patron goes to restaurant
patron sits down
patron recieves menu
patron orders
cook prepares meal
meal is served
patron eats
patron receives check
patron pays check
patron leaves restaurant

SCRIPT VARIABLES

how patron gets to restaurant
who gives patron menu
what patron eats
who serves the meal
who brings the check

When a script-based expectation has been violated, we can examine it in terms of script constants and script variables to see what aspects of the conceptualization are most interesting. Assuming that variations are more interesting than expectations, we can assign focus on the basis of script variables.

Given the question concept underlying Q1 (Did the waitress bring John a menu?) we examine this concept for script constants and variables. By accessing the restaurant script, we can determine that there is a script constant which corresponds to John getting a menu. Furthermore, there is a variable component within that constant act: the Actor. So focus is assigned to the Actor slot, and a Concept Completion question is generated by leaving the Actor slot unknown.

The same technique can be applied to Q2 (Did the waitress serve John a hot dog?). By examining the restaurant script we see that John being served is a script constant. What John is served and who serves him are script variables. So a Concept Completion question is generated by replacing the Actor and Object of the PTRANS with unknowns.

Q3 (Did John eat the hamburger) requires some additional processing. In this case the focus heuristic will send us looking for the answer to 'What did John eat?' But when we search the causal chain for a concept corresponding to John eating something, we can't find anything. Now a very strong expectation has been violated. John eating is a script constant. If there is nothing in the story representation corresponding to this script constant, then we must account for this unexpected omission. We must find out why John didn't eat anything. So to finally elaborate the answer to Q3 we must answer the question 'Why didn't John eat a hamburger?' Whenever a script constant is violated, we account for it by generating an Expectational question.

So we can now say when a Correction/Explanation elaboration requires a Concept Completion or an Expectational question. And in the case of Concept Completion we can determine which conceptual component should receive the focus and thereby determine which of all the possible questions will result in the most natural elaboration. In effect, we can find which secondary question will yield the most appropriate elaboration when the initial response to a Verification question is No. The establishment of focus in the original question was part of this task, and a script-based technique for focus establishment was invoked using the notion of script constants and script variables.

Figure 6 outlines the flow of control for a Verification question with the Correction/Explanation Option. The initial memory search tries to find the question concept in the causal chain representation. If an answer key is found with absolute credibility, the answer is Yes. If it is found with less than absolute credibility, the answer is Probably. If no answer key can be found, we try to generate a Concept Completion question by looking for a script constant with a script variable in the question concept. If we can identify a variable component within a constant act, a Concept Completion question is generated according to the script-based focus heuristic. If this Concept Completion question can be answered, we have our elaboration. (Did the waitress give John a menu? - NO. - Who gave John a menu? - THE HOSTESS GAVE JOHN A MENU.) If the question concept has a constant act but no variable component, an Expectational question is generated and answered. (Did John pay the check? - NO. - Why didn't John pay the check? - JOHN DISCOVERED HE HAD NO MONEY.) If a variable is found but the resulting Concept Completion question can't be answered, we generate an Expectational question for the final elaboration. (Did John eat a hamburger? - NO. - What did John eat? - JOHN DIDN'T EAT ANYTHING. - Why didn't John eat a hamburger? - THE HAMBURGER WAS BURNT.) The retrieval processing for these Expectational questions will be described in Chapter Seven.

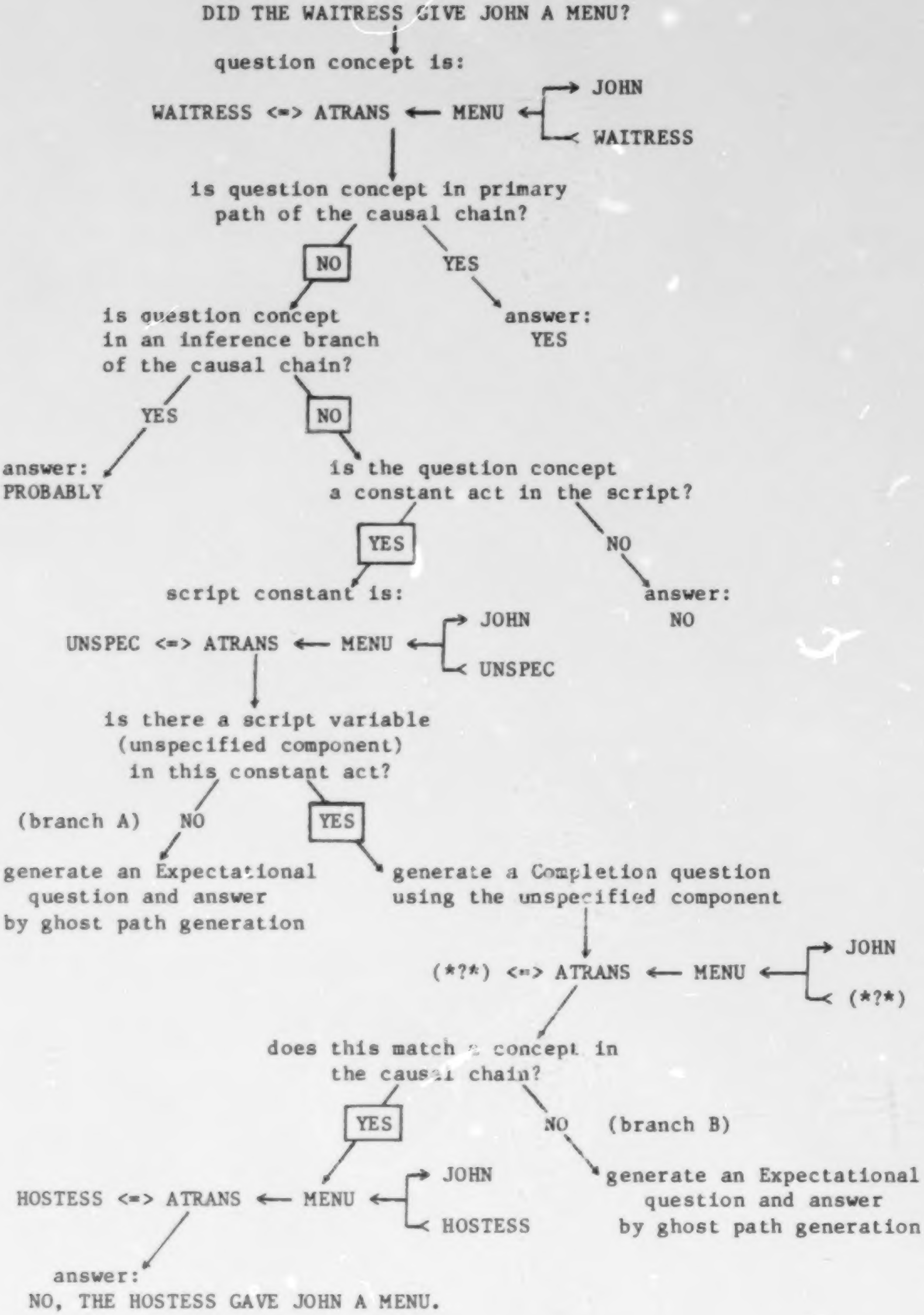

DID THE WAITRESS GIVE JOHN A MENU?
question concept is:
WAITRESS <=> ATRANS ← MENU
JOHN
WAITRESS
is question concept in primary path of the causal chain?
NO
YES
is question concept in an inference branch of the causal chain?
answer: YES
YES
NO
answer: PROBABLY
is the question concept a constant act in the script?
YES
NO
script constant is:
answer: NO
UNSPEC <=> ATRANS ← MENU
JOHN
UNSPEC
is there a script variable (unspecified component) in this constant act?
(branch A)
NO
YES
generate an Expectational question and answer by ghost path generation
generate a Completion question using the unspecified component
(*?*) <=> ATRANS ← MENU
JOHN
(*?*)
does this match a concept in the causal chain?
YES
NO
(branch B)
HOSTESS <=> ATRANS ← MENU
JOHN
HOSTESS
generate an Expectational question and answer by ghost path generation
answer: NO, THE HOSTESS GAVE JOHN A MENU.

Figure 6

FLOW OF CONTROL FOR VERIFICATION QUESTIONS

CHAPTER 7

UNDERSTANDING WHAT DIDN'T HAPPEN

Expectational (why-not) questions can be characterized by the fact that they ask about things which didn't happen. This poses a special processing problem in terms of memory retrieval. A story representation which contains information about events which occurred does not readily lend itself to questions about things which did not occur. It is also the case that some Expectational questions make more sense than others.

Q1: Why didn't Ford beat Carter?

is a perfectly reasonable question. But

Q2: Why hasn't Canada declared war against Mexico?

seems to be a silly question. It's as if some things which don't happen make more sense than other things which don't happen. How do we distinguish reasonable Expectational questions from unreasonable ones?

7.0 Introduction

In this chapter we will discuss Expectational questions and the processing required to answer them. Degrees of reasonableness in Expectational questions will be discussed and explained in terms of predictive knowledge structures. Finally, the retrieval heuristic used by SAM to answer Expectational questions will be described.

7.1 Aroused Expectations

When people read stories, their understanding process involves a predictive element which utilizes the expectations of the reader. Skillful writers are aware of the expectations they set up and they exploit these expectations in order to catch the reader off guard or dramatize important points. If you read that John just robbed a bank and the police are pursuing him in a car chase, you expect to hear some resolution. Either John will be captured, or killed, or he will outwit the police and escape. You expect to hear about some such outcome. You do not expect to hear about the results of John's last medical exam or whether or not John supports the Equal Rights Amendment. While this is an example where the reader may be conscious of being in suspense and wanting to find out what happens next, most of the predictions which occur during understanding occur on a much lower level and do not receive conscious attention.

Suppose you read that John wants to buy a sweater and he goes into a department store. In this case you have some expectations about what John is going to do next and what you might hear about next: he might look at a store directory, ask for information at a counter or information booth, he might go directly to the sweater department, or these things may be skipped over and you might hear about John looking at sweaters or buying one. These are relatively low level expectations which come from a department store script. They are not likely to create a sensation of suspense or strong interest. If a story goes on long enough merely setting up and confirming low level expectations, we are liable to feel bored by it.

All predictions made during the understanding process are made on the basis of knowledge about the world. A tremendous amount of knowledge is used in story understanding, including knowledge about stereotypic situations, activities, human relationships and motivations. People have a sense of what is normal and routine in the world and this prototypical knowledge is responsible for expectations aroused during understanding. Scripts and plans are knowledge structures which act as predictive mechanisms in order to generate expectations at the time of understanding.

7.2 Violated Expectations

When an expectation is not explicitly substantiated by the text, but is weakly confirmed by subsequent text, it can be incorporated into the memory representation as an inference. Inferences which are derived from expectations in this way are inferences made at the time of understanding and incorporated in the memory representation as explicit concepts in memory. But what comes of an expectation which was violated by subsequent text? There are only two possibilities: (1) The story representation could maintain a record of failed expectations in some manner, or (2) Missed predictions could be ignored in the construction of the story representation and therefore effectively forgotten.

There are certainly some violated expectations which must be incorporated into the story representation. If you read a murder mystery with a good twist to it, you might remember that you thought the butler did it and why you thought the butler did it until the last chapter. In fact the twist may be a predominant feature in your story representation. Long after you've forgotten the characters and the plot you might remember that the book led you down a clever garden path and surprised you at the end.

So there are some expectations aroused at the time of understanding which should be incorporated in the story representation. But should all expectations be recorded? The number of low level expectations which are violated in a short story can run into the hundreds very easily. It is hard to believe there is any good reason for preserving all such violated expectations in memory. Some criteria must be invoked in an effort to determine which are worthy of inclusion to memory and which are not.

This problem about expectations in the memory representation is just one aspect of a much larger issue. A theory of text comprehension must inevitably make some claims about what does and does not belong in the memory representation for a story. It is clear that there are times when a story representation should include some information about the understanding process which occurred at the time the story was read. Sometimes an expectation is aroused at the time of understanding which is later violated and we remember how we had made wrong assumptions. Is this a critical aspect of text comprehension? Would you say that someone did not comprehend a detective novel if they failed to recall how they had been misled by it? The question answering task is a good place to look for criteria whenever a problem arises concerning what does and does not belong in a story representation.

7.3 Answering Questions about Expectations

> John went to a restaurant and ordered a hot dog. But the waitress told him they didn't have any so he ordered a hamburger instead. When the hamburger came it was so burnt that John left.
>
> Q1: Why didn't John eat a hot dog?
> A1: Because the waitress told him they didn't have any.
>
> Q2: Why didn't John pay the check?
> A2: Because the hamburger was burnt.

These questions both ask about things which did not happen in the story. John did not eat a hot dog and he did not pay the check. The questions seem to be asking for the causality behind non-events. There is only one situation in which it makes sense to talk about the causality behind something that didn't happen: there must have been a time during understanding when there was an expectation that the act in question was going to occur. The question is then asking for the event or circumstance which interfered with that expectation. The stronger the expectation, the more sense the question makes.

If asked 'Why didn't John swim across the lake?' after reading the burnt hamburger story, the question makes no sense. We can't begin to answer it since we never had any expectations about John getting across a lake or going swimming. If asked 'Why didn't John order a salami sandwich?' the question makes more sense because there was an expectation that John would order something. Since there was no expectation that he should order a salami sandwich in particular, the question strikes us as being a little odd (why a salami sandwich?). But when asked 'Why didn't John eat a hot dog?' the question seems completely reasonable since we at one time expected John to eat a hot dog. As soon as the first sentence was read, we knew that John was a patron in a restaurant, he had decided that he wanted a hot dog, and he ordered a hot dog. Given this much information, we have a lot of low level expectations about what will happen next from the restaurant script. Knowing what normally happens in restaurants, we expect that the order will be communicated to a cook who will prepare the hot dog, then it will be served to John who will eat it, and he'll be given a check which he will then pay before he leaves the restaurant.

These expectations make it possible to skip a lot of intermediary information. 'John went to a restaurant and ordered a hot dog. John ate the hot dog quickly and left.' The second sentence follows the first smoothly only because the first set up expectations about what would happen next and these expectations included John eating a hot dog. In the burnt hamburger story, the use of the conjunction 'but' at the beginning of the second sentence is a warning device. It effectively tells us to watch out for something unexpected. But something can be unexpected only if we had expectations to the contrary. Since John eating a hot dog was an expectation aroused when John entered the restaurant and ordered, the question 'Why didn't John eat a hot dog?' is reasonable.

7.4 Answering Questions About Possibilities

> John's investigative report on the city court system was turning up some volatile political information. He had received threats in the mail but he didn't ease up on his story. One night a sniper fired at his bedroom window.

The processing for Expectational questions has not been implemented yet for plan-based stories. But when it is, we would like to produce answers like:

Q3: Why didn't John quit his story?
A3: He must have felt very committed to his work.

Q4: Why didn't John ask for police protection?
A4: I don't know. Maybe he didn't think it would help.

In this story we have some slightly less stereotypic expectations concerning the behavior of investigative reporters and the behavior of political people with shady dealings. When we hear that John has been threatened because of his work, we recognize that John is in a state of danger. While we don't know exactly what might happen, we do expect him to either comply with the threatening agent to remove the threat or to protect himself in some way. And so the questions about John quitting his story and asking for police protection are not unreasonable. It would make much less sense to ask 'Why didn't John buy municipal bonds?' At the time of understanding we have an expectation about the John getting hurt and this initiates some very general expectations about his behavior.

In the reporter story there were expectations about what was liable to happen in general. These expectations were not very specific in terms of exactly what would happen. We expected something might happen to John but we didn't know if he would get shot, or blown up, or if his home or family would be harmed. We expected John to do something about it but we didn't know if he would quit, seek more publicity, ask for police protection, or dig harder. Our expectations were in terms of general intentionalities and motivations rather than specific acts. Suppose we had asked:

Q5: Why didn't they threaten John over the phone?

Q6: Why didn't they shoot at his living room window?

Q5 and Q6 ask about specific actions which may have been feasible in the story but which did not take place. These questions are fundamentally different from Q3 and Q4 in terms of specificity. They relate to general predictions made at the time of understanding but they go into a deeper level of detail concerning specific actions which are consistent with these general expectations. It is harder to answer these questions since no expectations were made which relate to these acts specifically. One is tempted to answer 'Because the story just didn't go that way.' But if you feel compelled to give a more cooperative answer you try to reason out why the story took the turn it did instead of the one suggested by the question. So Q6 might be answered 'Maybe the light was on in the bedroom,' even though there is nothing explicit in the story supporting the choice of a target room.

A question that asks about an event which could have feasibly happened but which was not predicted at the time of understanding is asking about a possibility. The processing required to answer questions about possiblities is very different from the processing required when a question asks about a specific expectation aroused at the time of understanding. Questions about possibilities require inferences which were not made at the time of understanding. To answer why they didn't snipe at the living room window, we must first understand that this was an option they could have exercised in place of sniping at the bedroom window. Since no predictions were made at the time of understanding outlining all the possible ways they could have sniped at John, recognizing that this was an option requires an inference which was not made at any time previous to the time the question was asked. Once this recognition is made, we know the question makes a certain amount of sense. Had we asked 'Why didn't they offer John a season pass to the opera?' the question would fail to make sense since this is not recognizable as a reasonable plan of action for people who are trying to scare John off.

7.5 Classification of Expectational Questions

The last few sections have discussed very generally the ways an Expectational question can relate back to story representations. We classify Expectational questions into two general classes: (1) those which ask about specific expectations which were aroused at the time of understanding, and (2) those which ask about possibilities within a general expectation. If an Expectational question does not make sense, it either fails to reference a specific expectation aroused at the time of understanding, or it fails to specify a plausible option within a general expectation. Expectational questions asking about specific expectations are differentiated from Expectational questions about possiblities by the processes required to answer them. The remainder of this chapter will describe the processing of Expectational questions which ask about specific expectations aroused at the time of understanding.

7.6 Script-Based Expectations

When scripts are used in understanding stories they are applied to the understanding process in a strongly predictive manner [Cullingford 1975, 1976, 1977]. As soon as a script situation has been recognized, inferences are made concerning what has taken place and predictions are made about what is liable to happen next. Hearing that John went to a restaurant triggers the restaurant script which then makes predictions about John looking for a table, sitting down, getting a menu, deciding what to have, ordering, being served, eating, paying, and leaving the restaurant. If the next piece of text says that John ordered lobster, the previous predictions up to the ordering prediction become incorporated in the story representation as inferences about what must have happened between John going to the restaurant and John ordering. Inferences are made about John looking for a table, finding one, sitting down, getting a menu, and deciding what to eat[1].

Whenever a script is triggered in the understanding process, specific predictions are made on the basis of the stereotypic knowledge specific to the given script. Scripts are by definition knowledge structures of highly specific expectations. So a script-based prediction made at the time of text understanding will be a very specific conceptualization (e.g. the gas station attendant takes the cap off the gas tank, the clerk in the grocery store puts the items purchased in a bag, the waitress brings a check to the table). It may be the case that a story will deviate from the most routine path through a script.

Deviation from the default path of a script can occur when a script interference is encountered. In this event, earlier predictions made by the script applier become obsolete and must be updated by a new set of predictions. For example, 'John ordered a hamburger.' results in a path of predicted concepts taking John through the remainder of the restaurant script. Included in this path of predictions are conceptualizations for the cook preparing a hamburger, John being served the hamburger, and John eating the hamburger. But suppose the next input sentence is 'The waiter told John they didn't have hamburgers.' Now all the previous predictions about a hamburger are rendered obsolete. Having been told that the restaurant has no hamburgers, we no longer expect John to get one. The old predictions are therefore discarded and new predictions are loaded according to the expectations of the script. In this case, two paths of script instantiation are anticipated: John may reorder and carry on from there; alternatively John may decide to leave the restaurant.

---

[1] In actuality the predictions and inferences which would be made here are much more numerous. A few of the major ones are delineated here only to illustrate the inference processing.

Consider the story:

> John went to a restaurant and ordered a hamburger. But the waiter told him they didn't have any so John left.

The final story representation for this would be a causal chain containing conceptualizations for John entering a restaurant, sitting down, ordering a hamburger, being told there were none, and leaving. The story representation would have no record of the expectations which were present in the system just before the processing of the second sentence. These expectations were effectively 'forgotten' when the system revised its scriptal predictions in the course of understanding the second sentence. 'Forgotten' here means that no record of these predictions was entered into the story representation.

Now suppose we wanted to answer the question 'Why didn't John eat a hamburger?' This question only makes sense if we recognize that the concept of John eating a hamburger was an expected act at some point during understanding. But how can we recognize that the question makes sense if the story representation has no record of the failed expectations? This question cannot be answered unless the question concept can be related to the story representation as an act which would have taken place if only (something) hadn't happened. The question requires that we identify the one event in the story which wiped out our expectation that John would eat a hamburger. If we can identify the concept responsible for this revision in expectations, then we have an answer:

> Why didn't John eat a hamburger?
> Because the waiter said they didn't have any.

7.7 How to Remember Things You Forgot

While it is useful to know that answerable Expectational questions ask about failed expectations or possibilities which were alive at the time of understanding, we still have the problem of answering these questions on the basis of a story representation which has no history of failed expectations or alternative outcomes. We have described two classes of why-not questions: those asking about script-based expectations and those asking about plan-related possibilities. In this section we will describe a retrieval mechanism for answering why-not questions about script-based expectations. Given that a story representation does not contain all failed low-level expectations, a process is required which can reconstruct those failed expectations which were alive at some point during the understanding process.

7.7.1 Ghost Path Generation

If something is lost, it is reasonable to start looking for it at the place where it was last seen. The same idea holds when trying to recover a lost expectation. We must first go back to whenever it was that we last had the expectation. While we can't turn back time, we can pretend to go back in time by reconstructing processing states which occurred during understanding. The past processing states we are interested in are those which involve changes in the predictions current at that time.

When an expectation has been violated, it was last seen alive just before it was replaced by some other expectation. So if we are looking for lost expectations, it makes sense to look for them at those points in the story where there were shifts in the current predictions, where one set of predictions was replaced by another set. We need to be able to examine the story representation and find points in the story where predictions were revised.

When the knowledge structure responsible for comprehension is a script, there are some general script structures which reflect predictive shifts very simply. For example, script interference points are always places where script predictions are revised. A script interference is an event which is not normally expected in a smooth execution of the script, but which is encountered frequently enough to have stereotypic resolutions within that script. If John goes to a restaurant and is told he must wait 15 minutes for a table, he has encountered a script interference. Fifteen minute waits are not always assumed when one goes to a restaurant in general, but they occur often enough so that there are standard courses of action from which to choose when such an interference is encountered. John can go to another restaurant, he can stand and wait, he can go for a walk, have a drink at the bar, or slip the maitre-d' a tip. These are all stereotypic resolutions when one cannot be immediately seated at a restaurant.

Not all unexpected events are script interferences, even if they interfere with script execution. Suppose John goes to a restaurant, and when he orders the waitress ignores him, opens a Bible, and proceeds to read from the Book of Revelations. This is an unexpected occurrence from the point of view of the restaurant script and it interferes with the script. But since this occurrence is so completely removed from the restaurant script, the script cannot suggest what should be done in response to it. John will have to resort to plans in order to figure out how to handle the strange behavior of the waitress. If John encounters this situation very often he will incorporate it in his script as an interference point along with whatever resolutions he has learned to respond with. But most of us do not have this interference in our restaurant script.

Script interference points are important because they are places where script-generated predictions are revised. If John goes to a restaurant and orders a hamburger, the restaurant script predicts that his waiter/waitress will relay his order to the cook, the cook will prepare a hamburger, John will be served the hamburger, he will eat it, be billed for it, he will pay, and leave. But suppose John goes to a restaurant, orders a hamburger, and is informed that they don't serve hamburgers. Now we no longer expect his waiter/waitress to relay the hamburger order, we do not expect the cook to prepare a hamburger, John is not expected to get a hamburger, eat one, be billed for one, or pay for one. Being told that you can't have what you ordered is a point of interference in the restaurant script. Its standard resolutions are to order something else or leave. As soon as this script interference is encountered, expectations about what is going to happen change.

When the script applier is processing a story, it recognizes any interference points which are encountered. Interference points in a script are labeled as interferences. Their corresponding resolutions are also tagged accordingly. So when a script interference is encountered in a story, the script applier can easily tag it as such in the story representation. For example, suppose SAM reads the following story:

> John went to a restaurant and the hostess gave him a menu. When he ordered a hot dog the waitress said that they didn't have any. So John ordered a hamburger instead. But when the hamburger came, it was so burnt that John left.

The causal chain representation looks something like this: (in reality there are many more states and acts)

John enters restaurant

John is seated

John gets a menu from the hostess

John orders a hot dog

(I1) The waitress tells him thy don't have any

(R1) John orders a hamburger

Waitress serves John the hamburger

(I2) The hamburger is burnt →

John gets angry

(R2) John leaves the restaurant ←

In this story there are two interference/resolution pairs: the waitress telling John there are no hot dogs is resolved by ordering a hamburger and the hamburger being poorly prepared is resolved by leaving. These interferences and corresponding resolutions are tagged as such in the story representation.

When a why-not question is subsequently asked, we can easily identify those points in the story representation where expectations changed during understanding. We need only look for conceptualizations in the causal chain which are tagged as interferences. At each point of interference we know that a new set of expectations was generated by the script applier. The next problem is the reconstruction of expectations which were alive just before each point of interference. To reconstruct these expectations we must simulate to some extent the state of the script applier just before each interference was encountered.

To simulate states of the script applier, we ask the script applier to process another story. We want to know what the prediction queue of the script applier is when it understands the original story up to a point of interference but no further. To see this we will effectively ask the script applier to process the story over again, but this time we will only give it a truncated version of the story which cuts off just before an interference point.

To see what script predictions were alive just before the waitress told John there were no hot dogs, we will feed back to the script applier the causal chain from the story representation up to but not including the waitress telling him there are no hot dogs. The script applier is asked to understand this sequence of conceptualizations as a story. When it is done processing, it has a prediction queue and is ready to check the next input conceptualization against this queue. We have recaptured the state of the prediction queue as it was at the time of understanding just before the sentence describing the waitress' response to John's order was encountered.

Part of the prediction queue consists of a default path through the remainder of the script, instantiated according to what has been seen thus far. That is, the script applier has predicted what is liable to happen now if the execution of the restaurant script runs smoothly from now on without any more surprises. Given that John ordered a hot dog, the script applier predicts that the waitress will give that order to the cook, the cook will prepare the hot dog, the waitress will serve it to John, John will eat it, receive a check for it, pay for it, and leave. The script applier can generate this causal chain completing a default path through the script on the basis of predictions in its prediction queue. This script completion chain is called a ghost path. Each ghost path generated by the script applier is a causal chain along with a pointer to the place in the original story representation where the ghost path starts its branch from the actual story.

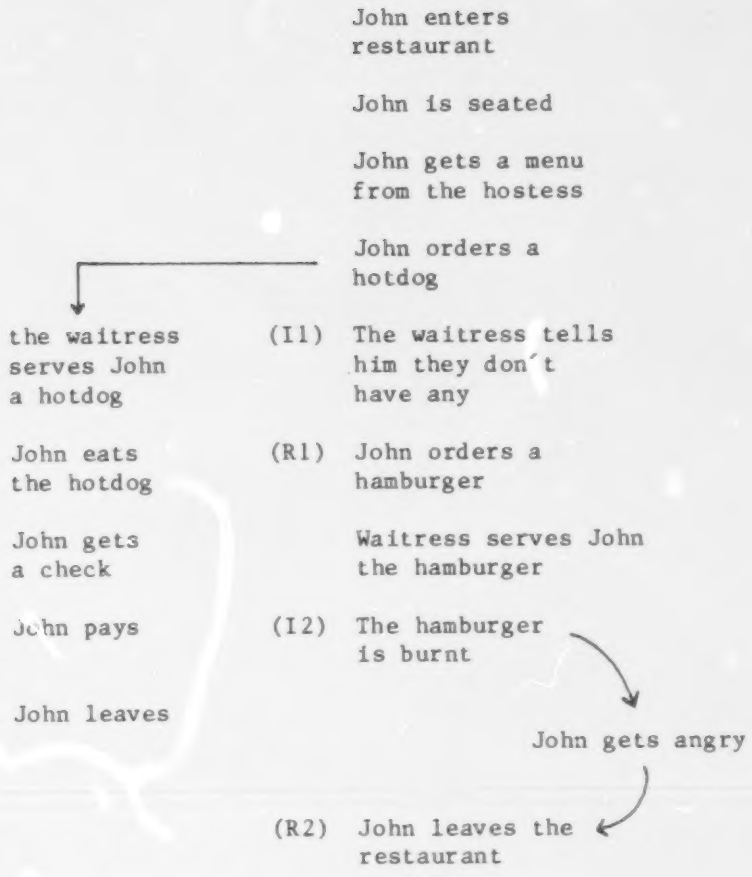

John enters restaurant
John is seated
John gets a menu from the hostess
John orders a hotdog
the waitress serves John a hotdog
(I1) The waitress tells him they don't have any
John eats the hotdog
(R1) John orders a hamburger
John gets a check
Waitress serves John the hamburger
John pays
(I2) The hamburger is burnt
John leaves
John gets angry
(R2) John leaves the restaurant

Figure 7

First Ghost Path

If we want to see the predictions which were alive just before the hamburger came back burnt, we go through the same procedure. The causal chain up to the point of the hamburger being burnt is handed to the script applier to be understood. When processing is finished, a ghost path which completes script instantiation is generated. This ghost path effectively recaptures expectations which were aroused at the time of story understanding but subsequently revised.

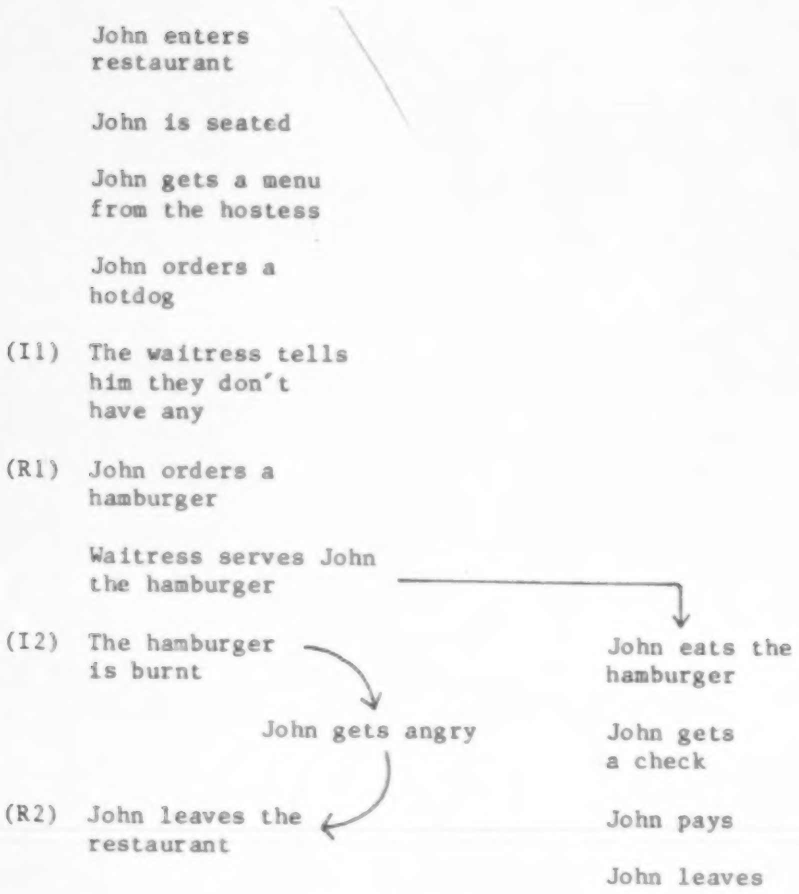

John enters restaurant
John is seated
John gets a menu from the hostess
John orders a hotdog
(I1) The waitress tells him they don't have any
(R1) John orders a hamburger
Waitress serves John the hamburger
(I2) The hamburger is burnt
John gets angry
(R2) John leaves the restaurant
John eats the hamburger
John gets a check
John pays
John leaves

Figure 8

Second Ghost Path

7.7.2 Using Ghost Paths

Once ghost paths have been generated, the processing needed to answer expectational why-not questions is fairly straightforward. Given a ghost path, we examine it for the question concept (non-negated). When the question concept is located, we follow the ghost path up to its origin where it branches off of the original story representation. Each such branch occurs immediately before a script interference. The answer to our Expectational question is that interference conceptualization. In this way we can answer:

Q1: Why didn't John eat a hot dog?
A1: Because the waitress told John they didn't have any hot dogs.

Q2: Why didn't John eat the hamburger?
A2: Because the hamburger was burnt.

In the event that a match can be made in more than one of the ghost paths, we trace the most recent path (the path whose branch from the story representation occurs closest to the end of the chain) back to the branching point and subsequent interference. So the question:

> Q3: Why didn't John pay the check?

would be answered:

> A3: Because the hamburger was burnt.

When SAM answers Q3 a slightly more involved answer can be produced:

> A4: John was angry because the hamburger was burnt and so he left.

This is due to the fact that paying the check is a pure script constant while eating a hamburger and eating a hot dog are acts which include script variables (see section 6.3 for a description of script constants and script variables). When a pure script constant has been violated, more explanation is needed than when acts with script variables are violated. If John didn't eat a hamburger, it is possible that he ate something else; the violation of the concept may be restricted to the instantiation of the variable component. But when John doesn't pay the check, there is no shift in a role instantiation which could explain why not. So when a constant act is encountered as the concept underlying a question, the retrieval heuristic is instructed to piece together in a causal template an answer including not only the interference (the hamburger was burnt), but its corresponding resolution (John left), and any intermediate mental state changes as well (John was angry). This retrieval heuristic is controlled by the Mental State Description Option of Content Specification (see section 4.2).

CHAPTER 8

FINDING THE BEST ANSWER

Some questions can be answered with many different answers, each of which is right and makes sense. When a memory representation contains information for many potential answers, retrieval is more complicated. The situation of multiple answers arises very easily. Even in stories which SAM can understand completely on the basis of scripts, we see how one event could be explained in terms of multiple causal antecedents:

Q: Why did John wash dishes at Leone's?
A1: Because he couldn't pay the check.
A2: Because he had no money.
A3: Because he was pickpocketed on the subway.

How is the memory search going to decide which answer should be returned? Should retrieval heuristics be aware of all the possible choices for an answer? Or should the memory search somehow 'know' where to look for the best answer and be oblivious to other possibilities? If a heuristic is aware of many answers, it must be able to make a choice. What rules can be formalized for finding the best answer? If a heuristic is blind to all but one answer, how can we be certain it will see the best one? In this chapter we will explore strategies for finding a best answer and discuss issues of memory representation which are related to this problem.

8.0 Introduction

Why-questions are difficult to answer because they often have many answers which are plausible, acceptable, and appropriate. Sometimes one answer is clearly better than another, and a good Q/A model should be able to discern which answers are better. While examining this problem, we will assume we are working with a system of cooperative and honest intentionality. Furthermore, we will restrict ourselves to the domain of story understanding where question answering is used to demonstrate comprehension. These assumptions will provide a useful base for a first pass at the problem. Given this orientation, we will say that the best answers are those which convey the most relevant information in the most efficient way.

One of the stories which SAM understands is the Leone's story:

> John went to New York by bus. On the bus he talked to an old lady. When he left the bus, he thanked the driver. He took the subway to Leone's. On the subway his pocket was picked. He got off the train and entered Leone's. He had some lasagna. When the check came, he discovered he couldn't pay. The management told him he would have to wash dishes. When he left, he caught a bus to New Haven.

After SAM has read this story, SAM answers:

> Q: Why did John wash dishes at Leone's?
> A: Because he had no money.

This is not a very good answer since it doesn't convey any information which the questioner couldn't have figured out for himself. A patron washing dishes in a restaurant is so scripty that being unable to pay the check is an immediate inference.[1] Given that John couldn't pay the check, it is obvious that John did not have any (or enough) money. So anyone who knows that John washed dishes at Leone's must also know by inference that John probably didn't have any money. In particular, the person asking Q1 must know this.

A much better answer would be 'John couldn't pay the check because his pocket was picked.' This answer conveys a lot of information. By inference it tells us that John had no money and therefore the management of the restaurant made him wash dishes. It communicates an entire causal chain which starts at the pickpocketing and ends with John washing dishes. And since part of the causal chain (John having no money and not being able to pay) is easily inferred, this answer has communicated information very efficiently.

Answer Selection Rule 1:

An answer which conveys information by inference is preferable to one which spells out such inferences explicitly.

---

[1] These inferences are 'immediate' and 'obvious' in the sense that people have no trouble making them. - The actual processes which produce these inferences are far from trivial or obvious.

In this chapter we will present an answer selection model which compares a number of possible answers to a question and selects one as the best answer. This model has been implemented in a computer program (ASP). After describing ASP, we will discuss some of the weaknesses inherent in the answer selection model. From there we will consider alternative approaches including strategies for knowledge state assessment and stronger structures for memory representation.

8.1 The Answer Selection Model

One approach to characterizing the best answer is to design a procedure which receives as input a set of possible answers to a question, and returns the best answer from that given set. Such a procedure will have to characterize the answers it examines along various dimensions and execute comparisons of answers in order to make a choice.

In trying to design such an answer selection procedure, we are forced to ask what elements of an answer are important. Then we must formalize those characteristics which intuitively seem to contribute to the strength or weakness of an answer. The model about to be described represents a first attempt in this direction. While it is not the case that this model will always pick the best answer from any set of possible answers, enough was learned from it to progress beyond the notion of answer selection.

8.1.1 Definitions

Data Context

The Data Context (DC) is the story representation to which a question refers.

Independent of Data Context

Answers to questions can be characterized as being Independent of Data Context (IDC) if they are answers one might reasonably guess without having read any story in connection with the question. These are answers people come up with when asked an artificial question out of the blue.

Q: Why did John see a doctor?
A: John was sick. (IDC)

Q: Why did John fall asleep?
A: John was tired. (IDC)

Explicitly Dependent

Answers which are derived from the Data Context by being explicitly present in the Data Context are Explicitly Dependent (ED) on the Data Context.

Implicitly Dependent

Answers which are derived from the Data Context by applying inference processes to the Data Context are Implicitly Dependent (ID) on the Data Context.

Examples:

EX1: One morning John noticed that his dog was having trouble walking. That afternoon he took it to the vet.

Q: Why did John take his dog to the vet?
A: It was sick or injured (IDC)
A: It was having trouble walking. (ED)
A: He wanted to make it well. (ID)

EX2: One day John broke the mainspring of his watch. The dentist who lived next door fixes old watches for a hobby. So John took his watch to the dentist.

Q: Why did John take his watch to the dentist?
A: He had broken the mainspring. (ED)
A: The dentist fixes old watches. (ED)
A: He wanted to get his watch fixed. (ID)

EX3: One day John broke the mainspring of his watch. The dentist who lived next door fixes old watches for a hobby. So John called the dentist.

Q: Why did John call the dentist?
A: He wanted to make an appointment. (IDC)
A: He had broken the mainspring of his watch. (ED)
A: The dentist fixes old watches. (ED)
A: He wanted to get his watch fixed. (ID)

EX4: One day John broke the mainspring of his watch. He took it to the jeweler's to see if they would buy it.

Q: Why did John take his watch to the jeweler's?
A: It was broken (IDC, ED)
A: He had broken the mainspring. (ED)
A: He wanted to see if they would buy it. (ED)

EX5: While walking home, John realized he didn't have his umbrella with him. He remembered last having it at the restaturant he went to for lunch. John walked over to the restaurant.

Q: Why did John go to the restaurant?
A: He wanted to get something to eat. (IDC)
A: He wanted to find his umbrella. (ID)
A: He last had his umbrella at the restaurant. (ED)

EX6: John got three F's on his report card. He decided not to show it to his parents.

Q: Why didn't John want to show his report card to his parents?
A: His grades weren't good enough. (IDC)
A: John was afraid they would be angry. (ID)
A: John had gotten three F's. (ED)

EX7: John was on the side of a highway when a large truck skidded off the road. At the scene of the accident John noticed an oil slick covering the pavement. When he saw a taxi approaching the spot at high speed, he waved his arms frantically, trying to signal the driver.

Q: Why was John waving at the taxi?
A: John wanted a ride. (IDC)
A: John wanted to prevent it from skidding off the road. (ID)
A: John wanted to stop it. (IDC)
A: John was trying to signal the driver. (ED)

8.1.2 More Definitions

The preceding definitions characterize single answers in relation to the text to which they refer. The following definitions characterize answers in terms of their conceptual content and relationships to other answers.

Causal Antecedent

Given two answers, A1 and A2, A1 is said to be a Causal Antecedent of A2 iff it makes sense to say 'A2 because A1.'

Intentional Consequent

Given two answers, A1 and A2, A2 is said to be an Intentional Consequent of A1 iff:

1) A2 is of the form 'X wanted to ... {C1} ...'
2) A1 is of the form 'X ... {C2} ...'
3) it makes sense to say
   'X ... {C2} ... in order to ... {C1} ...'

Plan Component

Given two answers, A1 and A2, A2 is a Plan Component of A1 iff:

1) A1 is of the form 'X wanted ... {C1} ...'
   or 'X needed ... {C1} ...'
2) It makes sense to say
   '... {A1} ... and X knew that ... {A2} ...'
3) X is not the actor of A2

Consistent

Two answers are said to be Consistent iff each can readily be inferred from the other in the given Data Context. For example, in EX5, John wanting to find his umbrella and John wanting to get something to eat are not Consistent answers. But in EX3, John's watch being broken and John wanting to get his watch fixed are Consistent answers.

Motive Oriented

An answer, A1, is said to be Motive Oriented iff

1) A1 is not of the form 'X wanted ...' and
2) A1 describes an activity or state which strictly precedes the activity or state of the question concept in time.

8.1.3 Selection Rules

The following rules are to be applied in succession. At the end of RULE 3, a tentative answer (TA) has been picked. The tentative answer may be changed by RULES 4-6. The tentative answer becomes the final answer only after the application of RULE 6 is completed.

RULE 1

An IDC is the preferred answer only if there are no ED's or ID's.

ILLUSTRATION OF RULE 1:

If we are told in a story that John always eats lasagna at Italian restaurants, and must then answer 'Why did John order lasagna at Leone's?' An IDC answer is 'Because he wanted to eat lasagna.' A much better response is the ED answer 'Because he always has lasagna at

Italian restaurants.'

RULE 2

An ID is preferred over ED's only when the question has at least one IDC and the ID is not consistent with the IDC's.

ILLUSTRATION OF RULE 2:

In EX 5, we have A1: Because he wanted to find his umbrella.
A2: Because he last had his umbrella at the restaurant.
A3: Because he wanted to get something to eat.

A3 is an IDC. A1 is an ID, and A2 is an ED. A1 is preferred since A1 and A3 are not consistent.

RULE 3

Given a choice of ED's, first eliminate those ED's which are also IDC's (these are the least interesting answers). Next test to see if there are any Motive Oriented ED's. Eliminate these. If there is still a choice, resort to RULE 3a.

ILLUSTRATION OF RULE 3:

In EX 3, we have ED1: Because he had broken the mainspring of his watch.
ED2: Because the dentist fixes old watches.

Since ED1 is motive oriented, ED2 is the preferred answer.

RULE 3a

Given a choice of ED's, first eliminate those ED's which are also IDC's. Next test each ED by removing its explicit concept from the Data Context.

The ED in question is the best answer if:

(a) The revised Data Context makes no sense, or

(b) The revised Data Context generates an ID which is not consistent with the ED in question.

ILLUSTRATION OF RULE 3a:

In EX4, we have ED1: Because he had broken the mainspring.
ED2: Because he wanted to see if they would buy it.

The revised DC with respect to ED1 is:
John took his watch to the jeweler's to see
if they would buy it.

This generates A1: Because it was broken. (IDC)
A2: To see if they would buy it. (ED)

neither (a) nor (b) of the test hold here.

The revised DC with respect to ED2 is:
One day John broke the mainspring of his
watch. He took it to the jeweler's.

This generates A1: Because it was broken. (IDC) (ED)
A2: Because he broke the mainspring. (ED)
A3: Because he wanted it fixed. (ID)

Here A3 is not consistent with ED2, (b) holds in this case, and so we choose ED2 as the answer.

ANOTHER ILLUSTRATION OF RULE 3a

In EX2, we have ED1: Because he had broken the mainspring.
ED2: Because the dentist fixes old watches.

The revised DC with respect to ED1 is:
The dentist next door fixes old watches as
a hobby. John took his watch to the dentist.

This generates A1: Because he had broken it. (ID)
A2: Because he wanted it fixed. (ID)

Neither (a) nor (b) hold here.

The revised DC with respect to ED2 is:
One day John broke the mainspring of his
watch. John took his watch to the dentist.

Since this DC makes no sense (why take a watch to a dentist?) (a) holds in this case, and so we take ED2 as our answer.

NOTE ON RULES 4-6: In the application of rules 4-6, we do not consider IDC's as possible replacements for the TA.

RULE 4

If the TA is the Causal Antecedent of another answer, replace the TA with the other answer.

ILLUSTRATION OF RULE 4:

In EX6 we have A1: John had gotten three on it.
A2: John was afraid they would be angry.

A1 is the TA by the time Rule 4 is applied. A1 is a causal antecedent of A2 since it makes sense to say 'John was afraid that they would be angry because he had gotten three F's on it. So A2 replaces A1 as the TA.

RULE 5

If another answer is the Intentional Consequent of the TA, replace the TA with the other answer.

ILLUSTRATION OF RULE 5:

In EX7 we have A1: John was trying to signal the driver.
A2: John wanted to prevent him from skidding off the road.

A1 is the TA. A2 is an intentional consequent of A1 since it makes sense to say 'John was trying to signal the driver in order to prevent him from skidding off the road. So A2 replaces A1 as the TA.

RULE 6

If another answer is a Plan Component of the TA, replace the TA with the other answer.

ILLUSTRATION OF RULE 6:

In EX3 we have A1: John wanted to get his watch fixed.
A2: The dentist fixes old watches.

A1 is the TA. A2 is a plan component of A1 since it makes sense to say 'John wanted to get his watch fixed and he knew that the dentist fixes old watches.' So A2 replaces A1 as the TA.

8.1.4 Implementing the Model

The answer selction program (ASP) which implements this answer selection model is not a fully automatic system. It is an interactive program which falls back on the user whenever an answer must be characterized in terms of the given definitions. For example, if the program needs to determine whether or not a given answer is Motive Oriented, it asks the user whether or not the state or activity of the question precedes the question concept in time. It then categorizes the answer according to the response given by the user.

This interactive approach was adopted in order to avoid constructing an actual language processing system. ASP does not understand the Data Context, the question, or the answers in any conceptual sense. It categorizes answers by querying the user and applies the selection rules without any understanding. A fully automatic implementation of the answer selection model would require tremendous amounts of world knowledge in order to recognize valid causal relationships, reasonable human motives, and common inferences.

The point of ASP is to identify what characterizations are important about competing answers to a question; not to solve all the problems of computational inference.

8.1.5 ASP Output

The next few pages contain three interactive sessions with ASP which illustrate how the program asks the user for information in order to apply the answer selection rules.

```
*ST1
(ONE MORNING JOHN NOTICED THAT HIS DOG  WAS  HAVING  TROUBLE  WALKING.
THAT AFTERNOON HE TOOK IT TO THE VET)
*
*Q1
(WHY DID JOHN TAKE HIS DOG TO THE VET?)
*
*A1
(IT WAS SICK OR INJURED)
*
*A2
(IT WAS HAVING TROUBLE WALKING)
*
*A3
(HE WANTED TO MAKE IT WELL)
*
*
*(ASP)

SET !A TO THE LIST OF POINTERS
*(SETQ !A @(A1 A2 A3))

NOW ASSIGN THE CATEGORY TYPES FOR EACH ANSWER IN !A
A1 HAS CATEGORIES: *(IDC)
A2 HAS CATEGORIES: *(ED)
A3 HAS CATEGORIES: *(ID)


(RULE 1 EXECUTED)

GIVEN THE DATA CONTEXT, CAN ONE OF 'HE WANTED TO MAKE IT WELL' AND 'IT
WAS  SICK  OR  INJURED'  BE INFERRED FROM THE OTHER?  (TYPE YES OR NO)
*YES


(RULE 2 EXECUTED: NO ID IS PREFERRED OVER THE ED'S)

THE TA IS: 'IT WAS HAVING TROUBLE WALKING'

DOES IT MAKE SENSE TO SAY HE WANTED TO MAKE IT  WELL  BECAUSE  IT  WAS
HAVING TROUBLE WALKING?  (TYPE YES OR NO) *YES


(RULE 4 HAS BEEN EXECUTED)

THE TA IS NOW: 'HE WANTED TO MAKE IT WELL'
```

```
(RULE 5 HAS BEEN EXECUTED)

THE TA IS NOW: 'HE WANTED TO MAKE IT WELL'

DOES IT MAKE SENSE TO SAY HE WANTED TO MAKE IT WELL AND HE  KNEW  THAT
IT WAS HAVING TROUBLE WALKING?  (TYPE YES OR NO) *YES


(RULE 6 HAS BEEN EXECUTED)


THE BEST ANSWER IS BECAUSE IT WAS HAVING TROUBLE WALKING





*ST2
(ONE DAY JOHN BROKE THE MAINSPRING OF  HIS  WATCH.   THE  DENTIST  WHO
LIVED  NEXT DOOR FIXES OLD WATCHES AS A HOBBY.  SO JOHN TOOK HIS WATCH
TO THE DENTIST)
*
*Q2
(WHY DID JOHN TAKE HIS WATCH TO THE DENTIST?)
*
*A4
(HE HAD BROKEN THE MAINSPRING)
*
*A5
(THE DENTIST FIXES OLD WATCHES)
*
*A6
(HE WANTED IT FIXED)
*
*
*(ASP)

SET !A TO THE LIST OF POINTERS
*(SETQ !A @(A4 A5 A6))

NOW ASSIGN THE CATEGORY TYPES FOR EACH ANSWER IN !A
A4 HAS CATEGORIES: *(ED)
A5 HAS CATEGORIES: *(ED)
A6 HAS CATEGORIES: *(ID)


(RULE 1 EXECUTED)


(RULE 2 EXECUTED: NO ID IS PREFERRED OVER THE ED'S)

(THERE IS A CHOICE OF ED'S: RULE 3 WILL BE EXECUTED)
(IDC'S ARE ELIMINATED FROM THE ED'S)
```

```
(THERE IS STILL A CHOICE OF ED'S)

DOES THE STATE OR ACTIVITY OF 'THE DENTIST FIXES OLD WATCHES' STRICTLY
PRECEDE THE STATE OR ACTIVITY OF THE QUESTION STATEMENT IN TIME?
(TYPE YES OR NO) *NO

DOES THE STATE OR ACTIVITY OF 'HE HAD BROKEN THE MAINSPRING' STRICTLY
PRECEDE THE STATE OR ACTIVITY OF THE QUESTION STATEMENT IN TIME?
(TYPE YES OR NO) *YES

(MOTIVE ORIENTED ED'S HAVE BEEN ELIMINATED)
(THERE IS STILL A CHOICE OF ED'S: RULE 3A WILL BE EXECUTED)

REVISE THE DATA CONTEXT BY REMOVING THE STATEMENT CORRESPONDING TO
'THE DENTIST FIXES OLD WATCHES'

DOES THE REVISED DATA CONTEXT STILL MAKE SENSE?
(TYPE YES OR NO) *NO


(RULE 3A HAS BEEN EXECUTED)

THE TA IS: 'THE DENTIST FIXES OLD WATCHES'

DOES IT MAKE SENSE TO SAY HE WANTED IT FIXED BECAUSE THE DENTIST FIXES
OLD WATCHES? (TYPE YES OR NO) *NO

DOES IT MAKE SENSE TO SAY HE HAD BROKEN THE MAINSPRING BECAUSE THE
DENTIST FIXES OLD WATCHES? (TYPE YES OR NO) *NO


(RULE 4 HAS BEEN EXECUTED)

THE TA IS NOW: 'THE DENTIST FIXES OLD WATCHES'


(RULE 5 HAS BEEN EXECUTED)

THE TA IS NOW: 'THE DENTIST FIXES OLD WATCHES'


(RULE 6 HAS BEEN EXECUTED)


THE BEST ANSWER IS BECAUSE THE DENTIST FIXES OLD WATCHES




*ST7
(JOHN WAS ON THE SIDE OF A HIGHWAY WHEN A LARGE TRUCK SKIDDED OFF THE
ROAD AT THE SCENE OF THE ACCIDENT JOHN NOTICED AN OIL SLICK COVERING
THE PAVEMENT WHEN HE SAW A TAXI APPROACHING THE SPOT AT HIGH SPEED HE
WAVED HIS ARMS FRANTICALLY TRYING TO SIGNAL THE DRIVER)
```

```
*
*Q7
(WHY WAS JOHN WAVING AT THE TAXI?)
*
*A20
(JOHN WANTED A RIDE)
*
*A21
(JOHN WANTED TO PREVENT IT FROM SKIDDING OFF THE ROAD)
*
*A22
(JOHN WANTED TO STOP IT)
*
*A23
(JOHN WAS TRYING TO SIGNAL THE DRIVER)
*
*
*(ASP)

SET !A TO THE LIST OF POINTERS
*(SETQ !A @(A20 A21 A22 A23))

NOW ASSIGN THE CATEGORY TYPES FOR EACH ANSWER IN !A
A20 HAS CATEGORIES: *(IDC)
A21 HAS CATEGORIES: *(ID)
A22 HAS CATEGORIES: *(IDC)
A23 HAS CATEGORIES: *(ED)


(RULE 1 EXECUTED)

GIVEN THE DATA CONTEXT, CAN ONE OF 'JOHN WANTED  TO  PREVENT  IT  FROM
SKIDDING  OFF  THE ROAD' AND 'JOHN WANTED TO STOP IT' BE INFERRED FROM
THE OTHER?  (TYPE YES OR NO) *YES

GIVEN THE DATA CONTEXT, CAN ONE OF 'JOHN WANTED  TO  PREVENT  IT  FROM
SKIDDING  OFF  THE ROAD' AND 'JOHN WANTED A RIDE' BE INFERRED FROM THE
OTHER?  (TYPE YES OR NO) *NO

THE TA IS:  'JOHN WANTED TO PREVENT IT FROM SKIDDING OFF THE ROAD'

DOES IT MAKE SENSE TO SAY JOHN WAS TRYING TO SIGNAL THE DRIVER BECAUSE
JOHN  WANTED  TO  PREVENT IT FROM SKIDDING OFF THE ROAD?  (TYPE YES OR
NO) *YES


(RULE 4 HAS BEEN EXECUTED)

THE TA IS NOW: 'JOHN WAS TRYING TO SIGNAL THE DRIVER'

DOES IT MAKE SENSE TO SAY JOHN WAS TRYING  TO  SIGNAL  THE  DRIVER  IN
ORDER TO PREVENT IT FROM SKIDDING OFF THE ROAD?  (TYPE YES OR NO) *YES
```

(RULE 5 HAS BEEN EXECUTED)

THE TA IS NOW: 'JOHN WANTED TO PREVENT IT FROM SKIDDING OFF THE ROAD'

(RULE 6 HAS BEEN EXECUTED)

THE BEST ANSWER IS BECAUSE JOHN WANTED TO PREVENT IT FROM SKIDDING OFF THE ROAD

8.2 Going Beyond Answer Selection

There are a lot of problems with the answer selection model described in the last section. It is easy to come up with examples where the model fails to find the best answer. By examining cases where the model fails, we can see what is missing and better understand what is needed.

8.2.1 What's Wrong with the Answer Selection Model

One fault derives from the fact that the model is not sensitive to preceding dialog. Each question and set of answers are considered in absolute isolation of a conversational context. Suppose the model decides that the best answer to

Q1: Why did John take his watch to the dentist?

is

A1: He knew that the dentist fixes watches for a hobby.

While this might be quite reasonable if Q1 is the first question in a dialog, it seems rather odd when Q1 is preceded by an exchange which establishes that John knew the dentist fixes watches for a hobby.

Q2: Did John know that the dentist fixes watches for a hobby?
A2: Yes.
Q3: Why did John take his watch to the dentist?
A3: He knew the dentist fixes watches for a hobby.

A more natural dialog would not contain repetitive information:

Q4: Did John know that the dentist fixes watches for a hobby?
A4: Yes.
Q5: Why did John take his watch to the dentist?
A5: He liked the dentist better than the jeweler.

If a question answering system is not sensitive to preceding dialog in some way, it cannot avoid exchanges which are needlessly repetitive. The basic problem seems to revolve around a very simple principle:

Answer Selection Rule 2:

A good answer does not tell the

questioner something he already knows.

The answer selection model described in the last section has no sense of what the questioner knows or doesn't know. In the above examples, the previous dialog provides information about what the questioner knows since can we assume that answers to previous questions are incorporated into the questioner's knowledge state. But does this mean that we that we have to formulate a system of knowledge state modeling before we can address problems in answer selection? It seems that some principles of retrieval can be identified without knowing what form a knowledge state model must assume.

When an answer to a question is dependent on information conveyed in previous dialog, we cannot talk about the retrieval processes which arrive at that answer without reference to a knowledge state model of some sort. But knowledge state assessment is important on the more immediate level of single questions and answers as well. When someone asks a question, they are telling you something about what is in their knowledge state. You have to know something in order to ask a question in the first place.

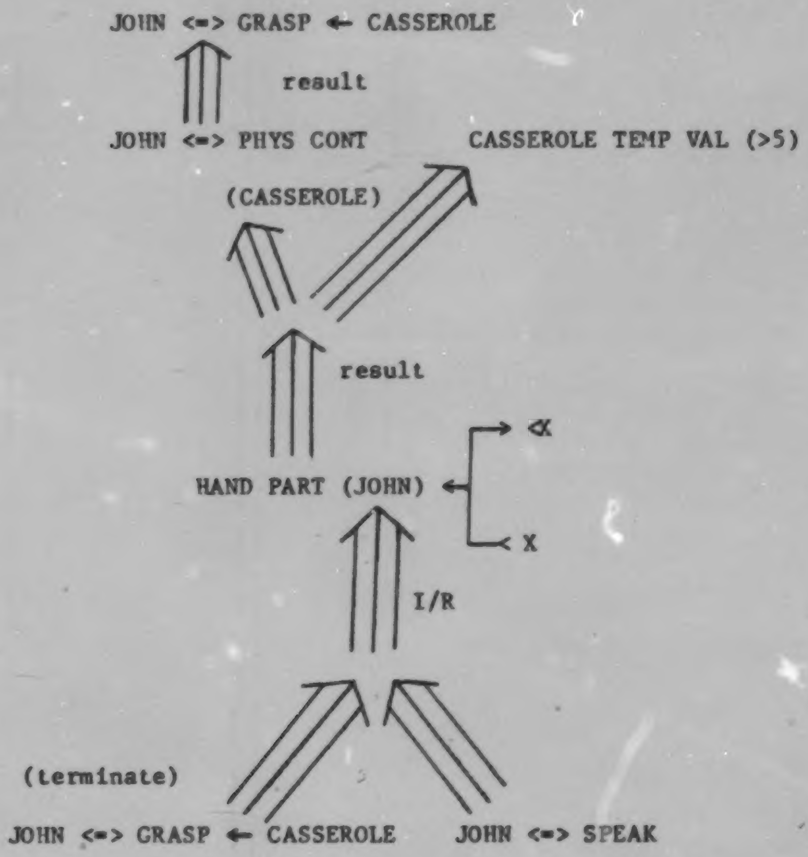
The answer selection model outlined in 8.1 does not examine questions with this in mind. In fact, the answer selection model doesn't look at the question very much at all. All of the selection rules are concerned with relationships between competing answers and relationships between answers and the data context. The only time the model looks at the question is when the answers are initially categorized as IDC, ID or ED answers. The identification of IDC answers involves the question because IDC answers are those which could be reasonably offered in response to a question out of the blue, without reference to any story. At some point in its processing, the model should examine the question for information about what the questioner knows in order to find the most appropriate answer.

If knowledge about what the questioner knows and doesn't know can be useful in forming an answer to a question, and if the question itself provides some information about what the questioner knows and doesn't know, there must be times when examining the question for such information is necessary in finding the best answer. In QUALM the notion of knowledge state assessment was mentioned but not examined very deeply (see section 3.1.3.2). The answer selection problem illustrates how some questions require more attention with respect to knowledge state assessment.

Consider the following story:

> John forgot the pot holders when he removed the casserole from the oven. When he picked it up, he yelled and dropped it on the floor.

In the causal chain representation of this story, both John yelling and John dropping the casserole share the same causal antecedent of John being burnt:


JOHN <=> GRASP ← CASSEROLE
result
JOHN <=> PHYS CONT
CASSEROLE TEMP VAL (>5)
(CASSEROLE)
result
<X
HAND PART (JOHN)
X
I/R
(terminate)
JOHN <=> GRASP ← CASSEROLE
JOHN <=> SPEAK

If the retrieval mechanism for why questions were designed to pick up the immediate causal antecedent of a question concept, both

Q4: Why did John yell?

and

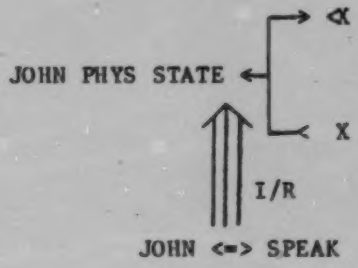
Q5: Why did John drop the casserole?

would be answered with 'Because he got burnt.' But something interesting happens when people are asked Q4 after reading the story and when (different) people are asked Q5 after reading the story. In an informal experiment, 11 people read the above story, 5 were asked Q4 and 6 were asked Q5. The responses were:

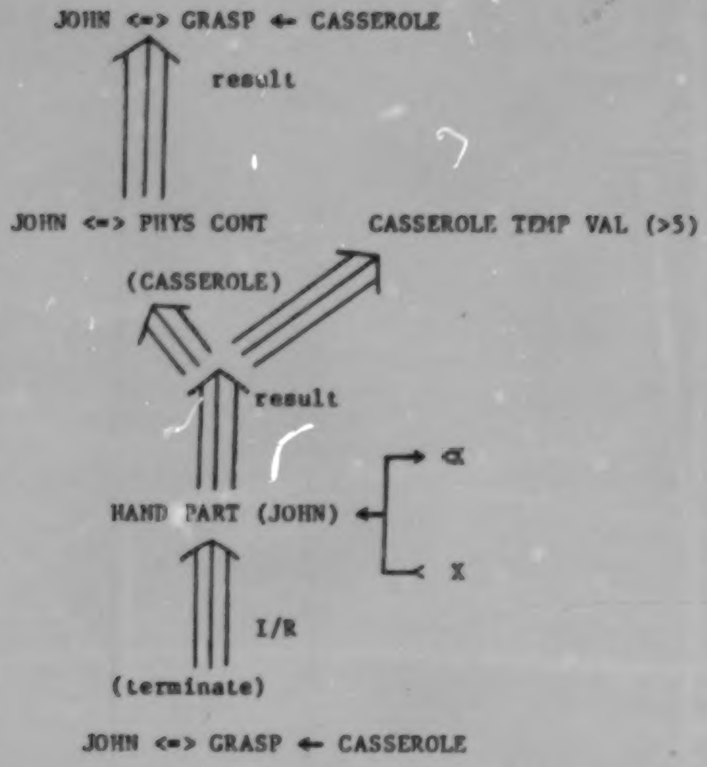
Q4: Why did John yell?
A4a: He burned his hand.
A4b: He had burned his hands on the hot casserole.
A4c: He hurt his hand picking up the hot casserole.
A4d: The hot casserole burned him.
A4e: Because he burned his hand.

Q5: Why did John drop the casserole?
A5a: 'Cause it was too hot to hold.
A5b: It was hot.
A5c: He burned his hand because it was hot.
A5d: Because it was hot.
A5e: Because it was hot.
A5f: It was hot.

In all the answers to Q4, a reference is made to the physical state change which John underwent. Everyone says that John got hurt or burnt. Some of the answers indicate that the casserole was responsible for it, but many of them simply say that John was burnt or John's hand was burnt without referencing the casserole at all. The answers to Q5 are somewhat different. Not all of the answers to Q5 mention that John was hurt or burnt. But all of the answers to Q5 explain that the casserole was hot. If one wanted to conceptualize what the questioner knew after getting an answer to Q4, the conceptualization would involve a negative physical state change initiating the reason for John yelling:



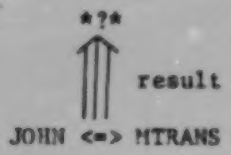
The conceptualization describing what the questioner would know after hearing an answer to Q5 would entail John picking up the casserole, John's physical contact with the casserole combined with the temperature of the casserole resulting in John getting burnt, and this initiating the reason for his dropping the casserole:


JOHN <=> GRASP ← CASSEROLE
result
JOHN <=> PHYS CONT
(CASSEROLE)
CASSEROLE TEMP VAL (>5)
result
α
HAND PART (JOHN)
X
I/R
(terminate)
JOHN <=> GRASP ← CASSEROLE

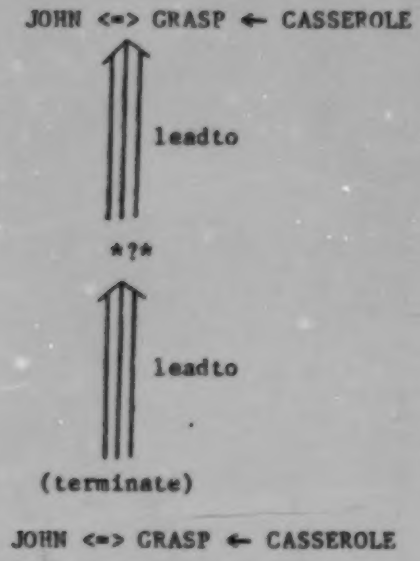
Some of the answers do not contain all of this explicitly. A5c is the only answer which describes John being burnt. But answers to Q5 leave the questioner with more information than answers to Q4 because the questioner starts out having more information in Q5 than Q4. The differences in answers are attributable to the differences in what the questioner knows at the time the question is asked. These differences are implicit in the questions.

Q4 indicates that the questioner knows John yelled. Q5 indicates that the questioner knows John had picked up a casserole (in order to drop something you have to have picked it up to begin with) and that John subsequently dropped the casserole.

The knowledge state corresponding to Q4 is:


?
result
JOHN <=> MTRANS

The knowledge state corresponding to Q5 is:



When a question asks about a single conceptualization, adequate answers can be found by looking for a single causal antecedent. But when a question asks about a causal chain, a good answer will be something which allows the questioner to construct the missing parts of the chain. This distinction is a major one from a processing point of view. Some questions ask about causal antecedents of single concepts and others ask about an entire causal chain. The process which returns a single antecedent is bound to be less involved than one which must deal with missing parts in a causal chain. To see exactly how much more involved, consider what happens when people answer questions of the second variety.

One way to make sure the questioner can fill in a causal chain is to spell it all out for him. So one answer to Q5 would be 'Because John got burnt from holding the hot casserole.' This answer explicitly describes John being in contact with the casserole, the casserole being hot, and John getting burnt as a result. But in answering Q5 nobody really does this. The answers to Q5 mention some parts of the chain and not others. The questioner is relied upon to fill in the missing parts by inference.

One inference that the questioner of Q5 is expected to make by himself is the fact that John is in physical contact with the casserole. This is a very low level inference which is immediate from John picking up the casserole. (The inference is immediate in the sense that people find it obvious - not in the sense that we have a good model of how it's done). Many of the answers to Q5 simply state that the casserole was hot. From this the questioner is expected to construct a chain including John being in contact with the casserole and consequently getting burnt.

When an answer must provide the questioner with enough information to complete a causal chain, the formation of that answer must use knowledge about what kinds of inferences can be readily made. People who answer Q5 with 'It (the casserole) was hot,' are assuming that anyone who knows

(1) John picked up the casserole
(2) the casserole was hot
(3) John dropped the casserole

can infer that

(1) John was in physical contact with the casserole
(2) John got burnt

This knowledge about inferences affects the retrieval processes which select an answer. How is this knowledge to be incorporated in the retrieval heuristics? There are at least two possible approaches:

(1) Possible answers could be generated and tested by a simulation program which would endeavor to simulate the questioner and see if a complete causal chain could be constructed on the basis of the proposed answer. This simulation might be achieved by implementing a causal-chain-filling program which used general rules of inference of the sort that Chuck Rieger proposed [Rieger 1975a].

(2) The memory representation and retrieval heuristics could be designed in some way which would enable the memory search to identify a good answer from which a chain could be constructed without simulating the construction itself.

Before discussing which of these approaches seems more promising we will review some of the important observations which have been made thus far.

*** A question must be examined to find out what the questioner does and doesn't know. This amounts to a form of MLOC assessment.

*** A question can be asking about either a single conceptualization or an incomplete causal chain. Knowledge state assessment based on the original question determines which case holds.

*** A question which asks about a single concept can be answered by describing a direct causal antecedent.

*** A question which asks about an incomplete causal chain should be answered by supplying only those parts of the chain which the questioner needs to complete the entire chain.

We have suggested two possible ways that an answer may be constructed to fill in an incomplete causal chain. The first method is a generation and test approach in which one bases memory retrieval on a simulation of the questioner processing proposed answers. The second method is to incorporate rules of knowledge state assessment within the retrieval heuristics. While principles of question-based knowledge state assessment and causal chain generation must be incorporated in the memory search in some way, it seems that this knowledge could be internalized in terms of the search techniques and memory representation alone. That is, a method of the second sort which encodes these principles on the level of memory representation and straight-forward retrieval heuristics would be a much more effective model. In the next section we will see how this can be done.

8.2.2 A Retrieval Rule Incorporating MLOC Assessment

In this section we will look at one way a retrieval heuristic can incorporate principles concerning knowledge state assessment and causal chain generation. With such heuristics the best answer to a why question can be found without considering competing answers. These retrieval heuristics must start with a question concept, analyze the question concept as it resides in the story representation, determine whether the question is asking about a single conceptualization or a causal chain, and in the case of an incomplete causal chain, find concepts which will enable the questioner to construct the entire chain.

KSA[2] RULE 1: Terminating Acts

This rule applies to Causal Antecedent questions which ask about the termination of a conceptual act or activity.

For example:

Q1: Why did John drop the casserole?
Q2: Why did John stop playing golf?
Q3: Why did John blow out the candle?

When a question asks about a termination, an immediate knowledge state assessment can be made. The questioner must know that the activity terminated was occurring over some interval of time preceding its termination. The person asking Q1 must know that John had picked up a casserole. In order to ask Q2 you have to know that John had been playing golf, and Q3 presupposed that the candle was burning.

Since these knowledge state assessments give the questioner knowledge of acts previous to the question concept, the question indicates that the first and last acts of a causal chain are known: the question is asking about what went on in the middle of that chain. So questions of this sort fall into the chain completion category.

---

[2] (KSA = Knowledge State Assessment)

The retrieval heuristic for KSA Rule 1 does not work for all questions which ask about terminating acts. But before we can further specify the questions for which this rule is intended, we must first refine our memory representation a bit. Up until now, our system of causal chains has functioned quite nicely with a very small number of rather general causal links (result, enable, reason, etc.) Here we see where it is necessary to refine the notion of a result link. In the casserole story GRASPing the casserole is a continual act. John's physical contact with the casserole is maintained simultaneously over the interval of time during which John is GRASPing the casserole. As soon as John ceases to GRASP the casserole, physical contact terminates. This relationship is a sort of simultaneous causality. In a paper by Chuck Rieger which identifies 28 types of causal relationships [Rieger 1975b], this relational link in which a result is continually dependent over an interval of time is called a continuous causality link. We will call it a continuous result link. This same type of link connects John being in physical contact with the casserole and John getting burnt. As long as John is touching the casserole, he is being burnt by it. Now we are ready to describe which questions our rule is intended for.

KSA Rule 1 applies to situations in which:

(1) The question concept describes a terminating act.

(2) Both the question concept and its corresponding initialization exist in the causal chain representation of the story.

(3) The initializing act has a causal consequent which is connected to it by a continuous result link.

This rule therefore applies to Q5 when it is asked in the context of the casserole story. John dropping the casserole is the termination of a GRASP act. Both John GRASPing the casserole and John terminating the GRASP are present in the causal chain representation. Finally, the act of John GRASPing the casserole has a causal consequent of John being in physical contact with the casserole which is connected by a continuous result link.

The retrieval heuristic for this situation is based on the following intuitions: When a conceptual act is ongoing over a period of time, it is very often the case that a causal chain is simultaneously maintained by the continuous act as long as the act persists. When this simultaneous chain leads to an undesirable state change, the initial act which maintains the chain is sometimes terminated in order to avoid the undesired state. The claim behind KSA Rule 1 is that when the three situational criteria of KSA Rule 1 are satisfied, the continuous act was terminated in order to avoid an undesired state change.

This is clearly the case in the casserole story. To see why the constraint concerning a continuous result link is necessary, consider the following story:

> John was shopping in a grocery store. Just as he was taking a steak from the meat cooler he noticed the ceiling begin to collapse in front of him. John dropped his basket and ran.

In this case John picks up a steak, is frightened by an impending disaster, drops the basket and runs away. There are no continuous result links in the causal chain representation for this story. Furthermore, one understands that John dropped the basket either because he was frightened or because he could run better without it. No one who understands this story would imagine that John dropped the basket because he thought it would stop the ceiling from collapsing. The shopping script assures us that the story representation will have the act of John picking up a basket in it. So the first two criteria of KSA Rule 1 are satisfied. The absence of a causal chain continuously dependent on John GRASPing the basket is what tells us not to assume that the basket was dropped in order to avoid the collapsing ceiling.

Given that the three criteria for KSA Rule 1 specify a situation in which the act is terminated in order to avoid some state, we still have the problem of deciding which concepts in the story representation should be returned for an answer. The answer should communicate enough information for the questioner to construct the causal chain. In the event that the causal chain has more than one or two intermediate concepts, a good answer will probably not spell out everything that is there. Some things can be left to the questioner to infer.

The problem is one of finding rules that can be applied to the memory representation to identify which concepts in the causal chain have to be mentioned. By looking at the answers people gave to Q4 and Q5, we see one immediate difference in retrieval. Some of the answers to Q4 (Why did John yell?) specified the immediate causal antecedent:

A4a: He burned his hand.
A4e: Because he burned his hand.

What is interesting is that when an answer to Q4 provides more information than this, no gaps are left in the causal chain. People do not tend to answer Q4 with an explanation like 'Because the casserole he was holding was hot.' That answer would require the questioner to infer that John was in physical contact with the casserole and that the casserole burnt him. All of the answers to Q4 mention John being burnt or hurt, the casserole being hot, and John being in contact with the casserole.

A4b: He burned his hands on the hot casserole.
A4c: He hurt his hand picking up the hot casserole.
A4d: The hot casserole burned him.

In A4d, physical contact with the casserole is not explicitly mentioned but it is implicitly present in the conceptual definition of 'to burn' which is being used here. This sense of the word requires an object$_1$ which is probably hot to come into contact with or close proximity to another object$_2$. This state change then results in a negative physical state change to object$_2$.

No one in our informal experiment answered Q4 with 'Because the casserole was hot.' This answer would require the questioner to piece together the casserole being hot with John yelling by making the inferences that John had touched the casserole and got burnt. Intuitively it seems that one could connect those concepts together with the right inferences, but apparently this is reaching too far. A good answer shouldn't make the questioner work too hard to make sense of it. The answers received in response to Q4 illustrate a general principle:

Answer Selection Rule 3:

When a question asks about a single conceptualization (as opposed to a causal chain), good answers do not require the questioner to infer the missing parts of an incomplete causal chain.

This principle leads us to a general rule about memory retrieval:

RETRIEVAL HEURISTIC:

> When a question indicates that the questioner has knowledge of only a single conceptualization, the answer should specify an immediate causal antecedent (in the case of a cooperative intentionality) or a complete causal chain leading up to the question concept (in the case of a more loquacious intentionality).

So questions about single concepts do not require retrieval heuristics which incorporate knowledge about what the questioner can infer. But we cannot avoid that problem when a question indicates that the questioner is trying to complete a causal chain. Q5 (Why did John drop the casserole?) is such a question. The answers received in response to Q5 all require the questioner to make some inference or inferences.

A5a: 'Cause it was too hot to hold.

This answer says that John dropped the casserole because the casserole was hot enough to make holding it result in some undesirable state. The answer does not specify what this state is. The questioner must infer that the undesirable state which results from holding a hot object is a negative physical state of being burnt. The answer is also constructed in such a way that it does not tell the questioner John was holding the casserole. So this answer requires the questioner to infer that John was in physical contact with the casserole, and that John was burnt by the casserole. With these inferences, and the concepts conveyed in the answer, the causal chain starting with John picking up the casserole and ending with John dropping it, is completed.

A5c: He burned his hand because it was hot.

This answer says that John dropped the casserole because he had burned his hand and that this was caused by the casserole being hot. In order to complete the chain this time the questioner must infer that John was in physical contact with the casserole. Everything else in the chain is given.

The four other responses simply described the casserole as being hot. This answer requires the questioner to infer that John was in physical contact with the casserole and that this resulted in John being burnt which in turn was responsible for John dropping the casserole.

While there are differences in these responses, there are some things which they all have in common. (1) They all explain that the casserole was hot. (2) None of them explain that John was in physical contact with the casserole. Intuitively some of this makes sense: knowing that someone has picked up an object drives an immediate low level inference that they must be in contact with that object unless you were specifically told that an intermediate device of some sort (tongs, gloves, a dust pan) was used. If John picks up the casserole we all expect him to be holding it for some interval of time immediately following the act of picking it up. The part that is a little less intuitive is that everyone describes the casserole as being hot. It appears that people are expected to infer that John was burned given that John was holding a hot casserole. But why is it the case that people are not expected to infer that the casserole was hot given that John was holding the casserole and he was burned? Why don't people always mention John getting burned instead of the casserole being hot? There are many possible explanations:

(1) It's easier to infer state changes than static properties. (Getting burned is a state change; being hot is a property.)

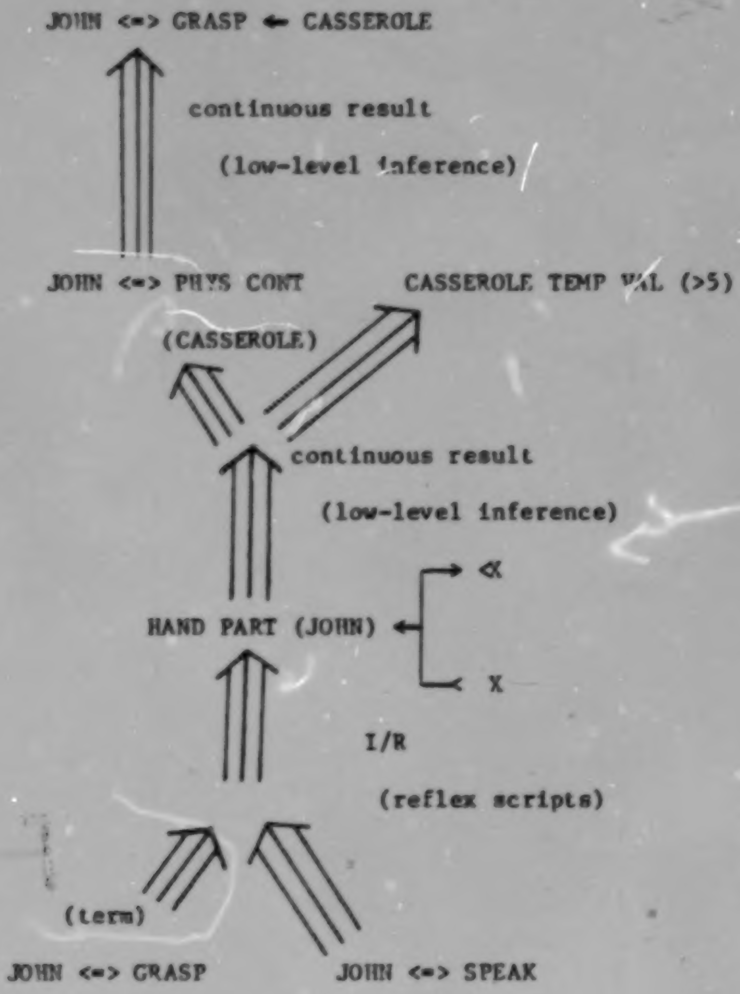
(2) It's easier to infer concepts which are simple linear pieces of a causal chain than it is to infer concepts which are causally concurrent in a causal chain. (Getting burnt has one causal antecedent and one causal consequent; the casserole being hot is simultaneously coupled with physical contact to cause John getting burnt.)

(3) It is harder to infer concepts which have fewer causal links tying them to the chain. (The casserole being hot is the only concept which is joined to the chain by a single link, a continuous result link; all the others have both a causal antecedent and a causal consequent.

There is even a fourth possibility which is not apparent from the causal chain representation as we have described it. It may be the case that concepts are chosen on the basis of information stored with each causal link. During the understanding process when the causal chain is being generated, each causal link is generated from some knowledge source. Some causalities are recognized by scripts, some by plans, perhaps some are derived from low-level inference mechanisms of the sort proposed by Rieger [Rieger 1975a]. It might be useful to differentiate links in terms of different structures within these knowledge sources.

We have actually done this already in the SAM system when the script applier tags interference/resolution pairs. Those tags provide additional information about what sort of general structure within a script was responsible for recognizing the causal relationships in chains beginning with interferences and ending in resolutions. All such chains are found in branches which leave the default path of the script. These tags are used in answering why questions which ask about script resolutions. It may very well be that the tagging of interference/resolution pairs is simply one particular case of an extensive system of processing tags which need to be incorporated in story representations. It will be easier to investigate this possibility when a system is designed which has access to both scripts and plans. When stories require different kinds of knowledge sources, it may be critical to have a story representation which easily reflects which knowledge sources were used to understand causalities.

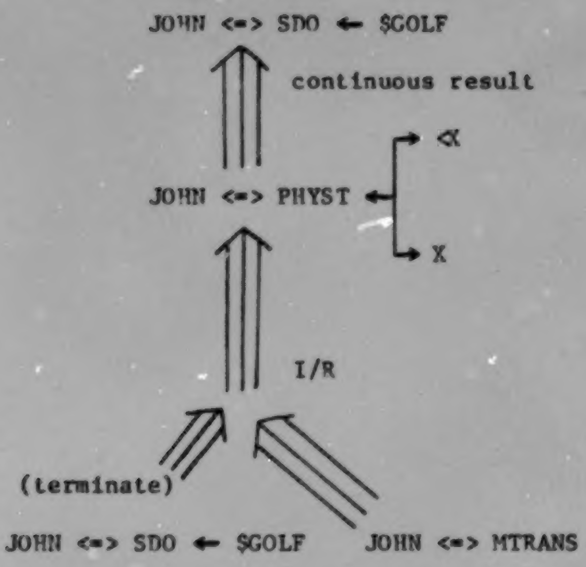
If such information were incorporated in the representation for the casserole story, it might look like:


JOHN <=> GRASP ← CASSEROLE
continuous result
(low-level inference)
JOHN <=> PHYS CONT
(CASSEROLE)
CASSEROLE TEMP VAL (>5)
continuous result
(low-level inference)
HAND PART (JOHN)
α
X
I/R
(reflex scripts)
(term)
JOHN <=> GRASP
JOHN <=> SPEAK

While it is very difficult at this time to guess which of the possible rules underlie retrieval, it seems plausible that rules of this general flavor can do the job. The idea of tagging causal links according to the knowledge sources responsible for them is a particularly promising approach. In fact, there is an example of memory retrieval falling under the control of KSA Rule 1 which seems to indicate a need for such tags very strongly. Consider a new story:

> John was playing golf one day when he pulled a muscle. He stopped playing, went home, and called the doctor.

The story representation for this looks like:



Now suppose we ask the question

Q6: Why did John stop playing golf?

This question in the context of the golf story satisfies the criteria of KSA Rule 1. The question asks about a termination, the story representation contains both the termination concept and its corresponding initialization (John playing golf), and the initializing concept has a causal consequent which is connected to it by a continuous result link. But the answers which people give in response to Q6 differ from those given for Q5 in an interesting way. Of five people who were asked to answer Q6 after reading the golf story, the responses given were:

A6a: Because he was hurt.
A6b: It hurt to play.
A6c: Because his muscle hurt.
A6d: He was worried it might get worse.
A6e: It was painful for him to play and he didn't
want it to get worse.

What is interesting about these responses in contrast to those given for Q5 is that two of these answers (A6b and A6e) refer to the fact that John was playing golf. None of the answers to Q5 mention John grasping the casserole. Yet the question Q6 indicates that the questioner knew John was playing golf just as Q5 indicated that the questioner knew John had picked up the casserole. So why tell the questioner something he already knows when answering Q6? The reason is fairly simple: while the questioner knows that John was playing golf, he does not have any information about what playing golf is related to causally. If an answer is given like A6a which simply states that John was hurt, it is harder to infer that playing golf was responsible for the hurt. It is not too difficult to infer it or people wouldn't give answers which require that causal inference (like A6a, A6c, and A6d). But it is harder to infer that playing golf results in being hurt than it is to infer that picking up a casserole results in being in physical contact with the casserole. It is harder because playing golf only occasionally results in being hurt but picking up an object always results in inferred contact with the object.

The story representation can reflect these fuzzy probabilities by including knowledge source tags on its causal links. If knowledge source tags were placed in the representation for the golf story, the link between playing golf and getting hurt would be attributed to an interference in the golfing script. The link from getting hurt to stopping would be from a resolution in the golfing script, and the link from getting hurt to calling the doctor would be from some plan or script concerned with the maintenance of good health. A retrieval heuristic that examined knowledge-source tags could be designed to return causal antecedents of script interferences but ignore causal antecedents of low-level inferences. Such an instruction would account for the fact that some people mention playing golf but nobody mentions picking up the casserole.

8.3 Looking Inside MBUILD's

In the last section a rule was developed which incorporated knowledge state assessment. The rule looked for structural aspects of a particular causal chain structure: (1) the initiation and termination of an act, and (2) a continuous result link emanating from the initiation of that act. Each of these features are structural aspects of causal chain representation.

In trying to develop a set of such structural retrieval rules, a very striking phenomenon reveals itself. Most answers to Why-questions describe motivations, goals, desires, or mental states.

Q1: Why did John take his watch to the dentist?
A1: He wanted to get it fixed.

Q2: Why didn't John show his parents his report card?
A2: He was afraid they would be angry.

Q3: Why was John waving at the truck?
A3: He wanted to prevent it from skidding off the road.

Q4: Why did John yell?
A4: He burned his hand.

All of these answers describe aspects of mental states. There is a difficulty in retrieving concepts of mental activity because causal chains do not provide much structure for such information. Consider the watch story:

> One day John broke the mainspring of his watch. His neighbor, a dentist, fixes watches for a hobby. John took his watch to the dentist.

In understanding this story, one makes roughly the following inferences about John's mental processes:

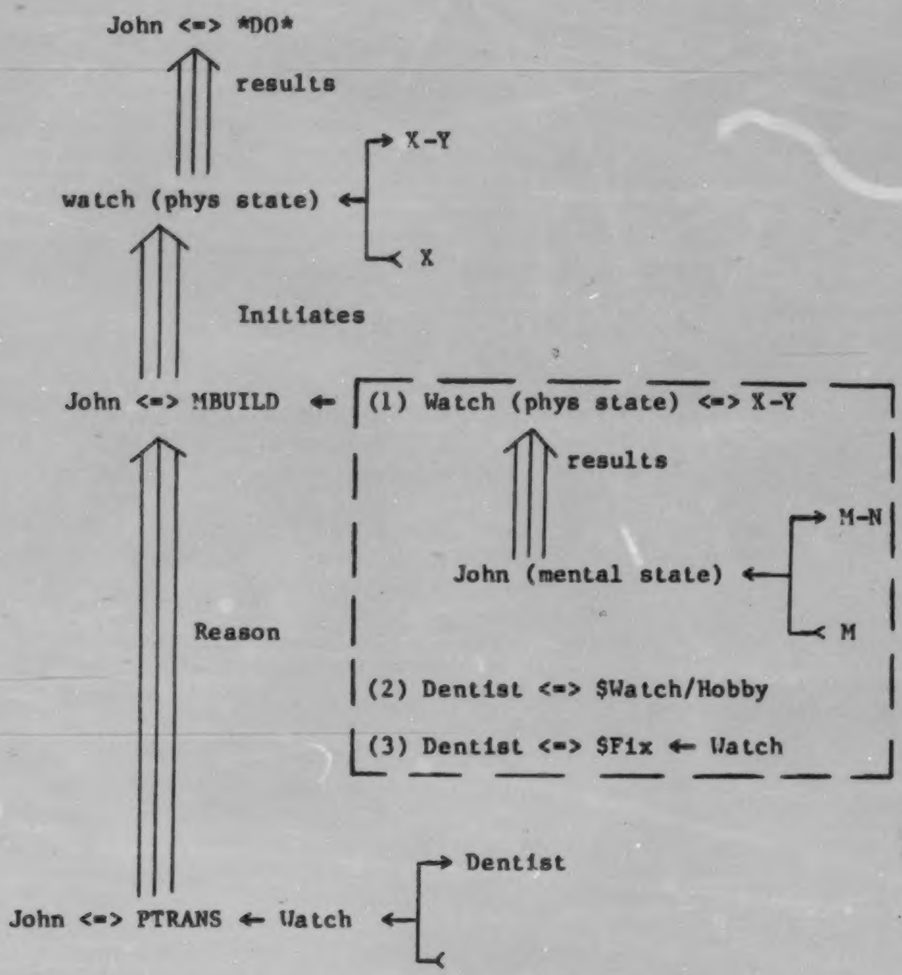
1) John liked his watch when it ran.
2) John's watch being broken makes John unhappy.
3) John didn't break his watch on purpose.
4) John wanted to get his watch fixed.
5) John knew that the dentist fixes watches for a hobby.
6) John thought that the dentist must get pleasure from fixing watches.
7) John thought that the dentist might fix his watch for the pleasure of it.

To see that each of these inferences is made, consider variations of the story which contradict each of these inferences. For example, to see that the second inference is valid, consider this variation:

> One day John broke the mainspring of his watch. John was very pleased by this. His neighbor, a dentist, fixes old watches for a hobby. John took his watch over to the dentist.

Here the contradiction forces us to think that (1) John is strange, or (2) John had been waiting for an excuse to visit his neighbor, or (3) John has some sort of plan in mind which we don't know about yet. None of these inferences were made in the context of the original story.

The problem with causal chains is that they cannot capture all of the inferences we would like to make in a story like this. The causal chain representation for the watch story looks like:


John <=> *DO*
results
X-Y
watch (phys state)
X
Initiates
John <=> MBUILD
(1) Watch (phys state) <=> X-Y
results
M-N
John (mental state)
M
Reason
(2) Dentist <=> $Watch/Hobby
(3) Dentist <=> $Fix ← Watch
Dentist
John <=> PTRANS ← Watch

All of the inferences about John's mental state belong inside John's MBUILD process. While the causal chain provides a place for all of the inferences about John's goals, desires, and mental states (within the MBUILD conceptualization), it does not specify how this information should appear. The problem we are faced with is what form these inferences inside John's MBUILD should assume.

Each of these mental activity inferences are related to each other by various kinds of causalities and mental strategies. Taking the watch to the dentist is a subgoal of wanting to get it fixed. The generation of this subgoal relied on knowing the dentist fixes watches for a hobby. The strategy for getting the dentist to fix his watch is to inform the dentist that the watch is broken. This strategy is expected to work because the dentist presumably enjoys fixing watches. John's ultimate goal (to get the watch fixed) resulted from the watch being broken, the broken watch displeasing John, and John seeing the revival of his watch as something which will alleviate his displeasure.

All of this should be present in the memory representation. But what kinds of links express these relationships? What is missing is a rigid structure for what goes on inside an MBUILD. If inferences about John's mental state were organized according to a fixed set of structural laws concerning mental processes, then retrieval rules based on this structure could be developed for retrieving information about goals and motives.

8.3.1 Plan-Based Retrieval Heuristics

Some problems are solved by invoking reliable scripts but others require cognitive processing beyond script selection. While it is clear that people often invoke scripts for restaurants or food preparation in response to their hunger states, it is not the case that a script can be invoked when you find out your wife is conspiring with your best friend to murder you. If we are going to answer questions about why people do the things they do, we have to understand the planning processes which are relied upon when there is no appropriate script for a given situation, or when a choice must be made from a variety of applicable scripts.

Suppose we were answering questions about the following story:

> John needed to finish his article for the magazine by Friday. He figured he had to work at least twenty hours a day to get it done, and even then it would be close to the wire. Everytime he became sleepy that week he felt angry and anxious. But he finished it on time. He came down with the flu right after the deadline and was very thankful to have gotten it then.

If we want to answer the questions,

> Q1: Why did John get angry when he became sleepy?
> Q2: Why was John thankful when he got the flu?

we must first understand that anger is not a common reaction to the daily need for sleep. In the same way we must realize that people do not normally experience relief when they get the flu. These questions cannot be answered without accessing information from the story which makes these causalities valid. Either the story representation or additional world knowledge must indicate that these causalities are not standard and need to be accounted for by special circumstances.

The answers to both of these questions must make some reference to John's overriding goal of making the deadline. His anger can only be understood as a response to goal interference. Needing to sleep meant taking time away from his work. This constituted an interference in his plan for making the deadline. Anger is an understandable reaction to goal interference. So an answer to Q1 must reference his threatened goal or the plan he invoked to achieve that goal:

> He was afraid he wouldn't make the deadline.
> (references goal)
>
> He needed to work almost continuously.
> (references goal plan)

Similarly, answers to Q2 must account for John's mental state in terms of his previous anxiety. The flu does not normally cause feelings of gratitude. But if we understand that the consequences of having the flu would have been much more serious had the flu struck a few days earlier, then we can understand that John is grateful for that difference of a few days:

> If he had gotten the flu earlier, he would have missed his deadline.
> (references alternative course of events and consequent goal failure)

The causal relationships between goals, goal interferences, plans, and mental state changes cannot be represented on the level of causal chains. Causal chains encode chronologies of physical events. When physical events are interpreted in terms of motivations and intentionality, we are concerned with another level of understanding which is founded on rules about people's desires and the plans they invoke to satisfy them. If we want to know why John was angry about becoming sleepy or why he was thankful to get the flu when he did, we must look inside John's head and understand what he was doing in terms of goals and plans. Where does this level of understanding belong in a story representation?

While all of this information can be appropriately placed inside John's MBUILD's, it may not be necessary to be quite so graphic about it. To put all of the planning information inside John's MBUILD's would entail placing a copy of the top level causal chain inside the MBUILD act which would appear within a syntax for internal MBUILDing. An alternative representation could avoid copying the causal chain acts by overlaying a system of plan-oriented links on the original causal chain. This new level of linkages would connect acts in terms of goal/subgoal relationships, goal/goal interference relationships, and other links which are not appropriate for causal chain syntax. This latter approach is the representational system adopted by PAM.

8.3.2 Retrieval Heuristics in PAM

PAM builds a causal chain of events just as SAM does, but final story representations generated by PAM also include pointers from conceptual acts to various planning structures such as goals, plans, subgoals, and goal interferences. This additional structure in the story representation is examined by the retrieval heuristics whenever a why-question is asked.

The only why-questions which have been implemented for PAM are those which retain their intitial conceptual categorization (Causal Antecedent or Goal Orientation) after interpretation is completed. For example, 'Why did John kill the dragon?' has the conceptualization representing 'John killed the dragon,' as its underlying question concept, and is understood to be a Causal Antecedent question.

The first task of the memory search is to find the question concept in the story representation. This is done by a straightforward matching search. Once the underlying question concept is found in the causal chain, its immediate causal antecedents are extracted from the causal chain of events. Each of these antecedents is then checked to see if it is tagged as an act which instantiates some plan. Any act which is tagged as being part of a plan instantiation will have a pointer to the immediate goal motivating that plan. If a plan instantiation is found, its immediate goal is returned as the conceptual answer. If the question type is Goal Orientation, the answer is expected to be found under the goal of a plan instantiation. If the question type is Causal Antecedent, there is one more place to look: when none of the direct antecedents yield a goal via plan instantiation, then the same list of antecedents is searched for a concept which was motivated by a state. If one is found, its causal motivation is returned as the conceptual answer.

In the dragon story, questions are answered both in terms of immediate goals and motivating states:

> John loved Mary but she didn't want to marry him. One day, a dragon stole Mary from the castle. John got on top of his horse and killed the dragon. Mary agreed to marry him. They lived happily ever after.
>
> Why did John get on his horse?
>
> Because he wanted to be near Mary.
> [Immediate Goal]
>
> Why did Mary agree to marry John?
>
> Because she was indebted to him.
> [Motivating State]
>
> Why did John kill the dragon?
>
> Because he wanted Mary to not die.
> [Immediate Goal]

The heuristics described above rely on four plan-oriented links which identify acts according to their relationships in terms of planned behavior.

CAUSEDBY - points to all direct causal antecedents
PLANINSTANTIATE - tags a conceptualization which instantiates a plan

FORGOAL - points to the goal behind a plan instantiation
MLEADTO[3] - connects a motivation to a conceptualization

This system of plan-based retrieval is a minimal beginning. It is far from what will be needed to answer questions like 'Why was John thankful when he got the flu?' Part of the difficulty in specifying a more complete set of retrieval heuristics derives from the fact that we don't know exactly what story representations should look like in many cases. A complete theory of predictive understanding mechanisms based on plans is still under development. We need to process a wider range of plan-based stories before we can identify crucial problems in retrieval and propose a complete set of retrieval heuristics for stories understood with plans.

8.4 Concluding Remarks

In this chapter we have outlined the complications involved in choosing a best answer and we have described various approaches to the problem. From our meanderings we will emerge with two basic conclusions:

(1) Some form of knowledge state assessment is needed. A capacity for knowledge state assessment can be implicitly encoded in the retrieval heuristics (8.2.2). But a comprehensive theory of knowledge state assessment will be needed regardless of how its rules manifest themselves in a Q/A system. In Chapter Ten we will look at some of the difficulties involved in building an explicit model of the questioner's knowledge state.

(2) Our memory representation must be strongly structured in the area of human motivations, goals, plans, and reasoning. Causal Antecedent and Goal Orientation questions often ask about human behavior. These questions rely on a thorough understanding of why people do the things they do.

In this chapter we have seen how critical it is for the story representation to encode information about structures internal to MBUILD's. In general, the Q/A task is very adept at detecting weaknesses in memory representations. In Chapter Nine we will see how Q/A problems have motivated a system of representational primitives.

---

[3] The MLEADTO link used here is not the same as the LEADTO link which acts as an unspecified causal link in causal chain syntax.

CHAPTER 9

CONCEPTUAL PRIMITIVES FOR PHYSICAL OBJECTS

When a class of questions is discovered which are particularly difficult to handle, it is probably because the memory representation is weak or inadequate. Some questions have proved to be difficult because our memory representations do not deal with physical objects very satisfactorily. Conceptual Dependency is a very action-oriented representation; its primitives are all Primitive Acts.

In this chapter a system of seven Object Primitives will be proposed. When nouns are represented by decomposition into these primitives we are able to see conceptual similarities and differences between objects in the same way that the Primitive Acts of Conceptual Dependency reflect conceptual similarities and differences between verbs. For example, a faucet and an underground spring are conceptually similar because they are both objects which commonly 'produce' water. The representation for these two concepts should reflect this similarity:

S1: John drank from the faucet.
Q1: What did John drink?
A1: Water.

S2: John filled his canteen at the spring.
Q2: What did John get at the spring?
A2: Water.

When we hear S1 we infer that water came out of the faucet. A similar inference is made for S2 about water coming out of the spring. Conceptual descriptions of objects should encode the knowledge needed for inferences of this sort. Furthermore, the organization of this knowledge should be structured so that the same inference mechanism which produces A1 will also produce A2. If inferences about different objects rely on different mechanisms for each object, we do not have a very viable theory of inference. Processes of inference should depend on conceptual descriptions of objects, but not on specific objects themselves. The Object Primitive descriptions for a faucet and a spring make it possible for a single inference mechanism to work in both cases. Object Primitive decompostitions provide a representational system for encoding knowledge about objects needed for natural language processing.

9.0 Introduction

Many inferences rely on knowledge about physical objects and a mundane understanding of physical causality. For example, if John is drinking coffee in the back of a car, and the coffee spills when the car hits a bump, we should infer that the coffee spilled because the car hit the bump. But if we were told instead that the coffee spilled when the car radio went dead, we would not want to infer that the car radio going dead was responsible for the coffee spilling. These inferences cannot be made on the basis of scripts. If we put into a riding-in-the-car script all of the possible events which would result in liquid spilling from an open container, the script would contain too much information to be workable. Even if all that information were successfully incorporated in a car script, it would have to be duplicated for a train script, and then a bus script, etc. etc.

People have a mundane knowledge of physical causality which they use in their inferencing processes. Knowledge about when liquids can spill must be general, and not tied to all the specific situations in which liquids can spill (in cars, trains, boats, airplanes, etc.). In order to build a natural language processing system, a knowledge base incorporating this mundane knowledge of the physical world must be developed.

One general inference mechanism should be able to recognize that a full coffee cup can spill during a bumpy car ride for the same reasons that can cause a full glass of water to spill in a dining car when the train pulls out. But a general rule of this sort requires a representational system which recognizes how cars are similar to trains and how full glasses of water are similar to full coffee cups. A conceptual representation for objects is required for very fundamental recognition processes.

Yorick Wilks [Wilks 1976] pointed out a critical problem in the current formulation of script application which illustrates the recognition problem nicely. Wilks refers to a paper by Eugene Charniak [Charniak 1975a] in which Charniak's formulation for a supermarket frame (script) includes references to entities like 'basket.' For example, the script will contain an act describing the shopper obtaining a basket. Wilks asks (quite rightly), how can such a surface representation be implemented in an understanding system where sentences must be recognized which describe the shopper getting a box, carrier-bag, push-cart, plastic sack, or any other number of possible carrying devices? How will the system understand that these various lexical items refer to objects which should match the notion of a 'basket' in the supermarket script? This criticism is just and deserving of attention. Wilks could have been talking about SAM's restaurant script as well. But Wilks goes on to cite this failing as a weakness inherent in the notion of scripts, when in fact, this failing merely indicates that we do not have an adequate representation for physical objects. At the end of this chapter we will return to this problem of script application with a solution based on a new representational system.

The representational system we shall propose is designed to facilitate inference processes. For example, in understanding, 'John went to the store and got some milk,' it is important to infer that the milk is in a milk container. No one hearing that sentence imagines John cupping a small quantity of milk in his hands. But, 'John went into a fancy restaurant and got some milk,' should be understood to mean that John ordered some milk and received it in a glass. People make inferences of this sort as soon as they hear these sentences. If these inferences were not made, then the following stories would not be bothersome:

> S3: John went to the store and got some milk. But the glass was very full and he spilled it on the floor.
>
> S4: John went into a fancy restaurant and got some milk. But the container was partially open so he asked for another one.

In S3 the reference to a very full glass which spills doesn't make sense. Stores don't supply milk in open glasses. In S4 the same sort of confusion arises. We expect John to be served milk in a glass. When we hear that it came in a container we are forced to conclude that the restaurant must be more like a cafeteria than the term 'fancy restaurant' suggests.

9.1 Object Primitives

In the same way that a set of Primitive Acts have been developed in Conceptual Dependency for the conceptual representation of acts, a set of Primitive Objects are used to represent physical objects. Seven primitives will be proposed for representing physical objects. Just as a verb is conceptually represented by a decomposition into Primitive Acts, a noun is conceptually represented by a decomposition into Object Primitives. When two objects are conceptually similar in some ways and different in others, their Object Primitive decomposition should reflect those similarities and differences. Object Primitive decomposition is very reminiscent of decomposition into Primitive Acts, but with one exception. When an object is decomposed into an Object Primitive description, it is usually described in terms of simultaneous and parallel Object Primitives. This is somewhat different from the primitive decompostion of actions, where each verb receives only one 'top-level' Primitive Act in its decomposition[1].

---

[1] E.g. 'give' is usually a top-level ATRANS with an instrumental PTRANS. Multiple Object Primitives are not embedded within a single hierarchical structure.

The primitives proposed here are designed to encode prototypical information about objects. They reflect normal expectations which people have about familiar objects. But 'normality' is a property which changes as context changes. While milk is normally found in a milk container in a store, it is normally found in a glass when being served at a restaurant. Therefore, Object Primitives rely to some extent on contingent descriptions. That is, part of the information in an Object Primitive decomposition may be applicable at some times and not at others. For example, an Object Primitive description of milk will encode the expectation that a shopper in a grocery store will find milk in a milk container, but a restaurant patron expects to find milk in a glass.

9.1.1 SETTING

When an object is big enough to hold a person it is often called a place. From a conceptual viewpoint, what is important about a place are the activities which are predictably associated with that place. Places like dining cars and grocery stores are characterized by the scripts associated with them. When a place has associated scripts, it is represented by the Object Primitive SETTING. SETTINGs can be associated with other SETTINGs as well as scripts. For example, a dining car is a SETTING which invokes the related SETTING of a passenger train. Once we hear that John is in a dining car, we immediately infer that John is in a passenger (coach) train. If we hear a subsequent reference to 'the train' there is no confusion about what train:

> John was sitting in a dining car. The train pulled out.

When the second sentence references a train there is no ambiguity about what kind of train. We do not wonder if it is freight train, a toy train, a subway train, or a train of thought. We have already been told (by implicit inference) that John is on a passenger train. The Object Primitive SETTING describes objects in terms of their associated scripts and other SETTINGs[2]:

```
[WashingMachine

  (a SETTING with

   <Scripts = $WashingMachine>

   <Settings = Home, Laundry>)]
```

---

[2] It is sometimes useful to describe a SETTING in terms of a single conceptual act as well. In these cases the slot name Acts is used. For example, a sandwich has a SETTING description with Act = INGEST. We will see an example of how associated acts are used in section 9.2.3.

```
[Dining Car
  (a SETTING with
   <Scripts = $Restaurant, $Preparefood>
   <Settings = Passenger Train>)]

[Classroom
  (a SETTING with
   <Scripts = $Teaching>
   <Settings = School>)]

[Pen
  (a SETTING with
   <Scripts = $Pen>)]
```

A SETTING does not need to be a place; it is either a place with situational scripts or an object with instrumental scripts. When a SETTING is a place (e.g. a dining car or classroom) the associated scripts describe activities which take place within those SETTINGs. But when a SETTING is not a place in the sense of being occupied by people (e.g. a washing machine or pen) then the associated scripts describe the instrumental or functional aspects of that object. Since the normal use of an object can be described in terms of a script, there is no problem encoding such information under the primitive SETTING. SETTINGs do not distinguish between places where scripts take place and objects which are the primary prop of a script. The scripts themselves make this distinction. A situational script describes an activity which commonly occurs in a fixed situation, while an instrumental script describes an activity which can occur in various situations but which requires a specific prop for its execution. Hence the restaurant script depends on a restaurant locale, but the pen script relies on only the presence of a pen.

9.1.2 GESTALT

Many objects are characterized as being something greater than the sum of their parts. Trains, stereos, universities, kitchens, all evoke images of many components which may interrelate and interact in any number of ways. The Object Primitive which is designed to capture these clustering effects is a GESTALT. All GESTALT objects are described by their Parts. A place setting is a GESTALT object whose Parts include a plate, knife, fork, spoon, glass, napkin, placemat, and so forth. But a place setting is more than just the set of those elements. It is a particular configuration of those elements. If a plate is balanced on an upside-down glass, we would hesitate to recognize that as part of a place setting. Similarly, a train is thought to be a linear configuration of cars usually headed by the engine; it is not a set of cars piled up on top of each other.

```
[Freight Train

  (a GESTALT with

   <Parts = Engine car, Freight Cars, Caboose>

   <Configuration = Linear string of Engine,

                    freight cars, & Caboose>)]


[Place Setting

  (a GESTALT with

   <Parts = Plate, glass, bowl, silverware, napkin>

   <Configuration = Radial configuration with plate

                    at center, silverware at right

                    on top of napkin, glass at one

                    o'clock, bowl at eleven o'clock>)]
```

9.1.3 RELATIONAL

In addition to objects which are described by their components or parts, many objects are described by the relationships they normally assume with other objects. Containers (rooms, bottles, shopping carts, etc.) are described by a capacity for containment. Supporting objects (tables, chairs, plates, etc.) are described by their ability to support other things. Hinges support doors, bulletin boards hold papers, and blackboards can be covered with chalk. The Object Primitive which encodes prototypical relationships between objects is a RELATIONAL object.

All RELATIONAL objects have a Relationlink which specifies the relation which the object normally assumes. The Relationlink value for a table will be 'on-top-of' while a bottle takes the Relationlink value 'inside-of'. Each RELATIONAL object includes constraints for the relations specified under its Relationlink. For example, a piano bench of the variety which opens up has two Relationlinks: on-top-of and inside-of. But different constraints operate on the objects which assume these relations. Something which goes inside the piano bench must not exceed certain dimensional constraints. A final aspect to RELATIONAL objects is the possiblity of instrumental objects which enable certain relationships. For example, a bulletin board maintains a 'stuck-to' relationship with papers but only if a thumbtack is utilized.

```
[Table
 (a RELATIONAL with
  <Relationlink = on-top-of>)]

[Blackboard
 (a RELATIONAL with
  <Relationlink = stuck-to>
  <Constraints = chalk>)]

[Bulletin Board
 (a RELATIONAL with
  <Relationlink = stuck-to>
  <Constraints = paper>
  <Instruments = thumbtack>)]
```

One aspect of RELATIONAL objects which may seem to be a natural part of these descriptions is the notion of specific defaults. For example, an egg carton is a RELATIONAL object with a very strong tendency to harbor eggs. The idea of default objects of this sort is a very strong one which enables a large class of inferences. If John fills his lighter, it is standard to infer that the lighter was filled with lighter fluid. If a pen leaks, it is expected to leak ink. When the tank of a car is full, it should be full of gasoline. This characteristic is not restricted to RELATIONAL objects, however. A faucet is expected to produce water and a radio commonly emits music or verbal communication. The ideas of production and consumption among objects motivate the next two Object Primitives.

9.1.4 SOURCE and CONSUMER

A SOURCE is an object which is characterized by its tendency to produce other things. Sugar bowls are SOURCEs of sugar, egg cartons are SOURCEs of eggs, and faucets are SOURCEs of water. A CONSUMER is an object which tends to consume other things. A drain consumes liquids and a slot machine consumes coins. Of course a slot machine can also be a SOURCE of coins but it tends on the average to be more of a CONSUMER.

A SOURCE is related to the objects it produces by an Output link while CONSUMERs have corresponding Input links. In addition to these descriptors, some SOURCEs and CONSUMERs require Activation scripts and/or Deactivation scripts. For example, a radio or a light fixture are both sources which need to be activated and deactivated.

```
[Wine Bottle
  (a SOURCE with
   <Output = Wine>
   <Activation = $Pour>)]

[Book
  (a SOURCE with
   <Output = MObject>
   <Activation = $Read>)]

[Mailbox
  (a CONSUMER with
   <Input = Letters>)]

[Ice Cube Tray
  (a SOURCE with
   <Output = Ice Cubes>)
  (a CONSUMER with
   <Input = Water>)]

[Sponge
  (a SOURCE with
   <Output = Liquids>
   <Activation = $Squeeze>)
  (a CONSUMER with
   <Input = Liquids>
   <Activation = $Wipe>)]
```

```
[Pipe
  (a SOURCE with
   <Output = Smoke>
   <Activate = $Smoke>)
  (a CONSUMER with
   <Input = Tobacco>
   <Activation = $Smoke>)]
```

Object Primitives can be used to encode conceptual senses of some verbs and adjectives as well as nouns. For example, 'to empty' something means to use it as a SOURCE. 'To fill' something means to use that thing as a CONSUMER. So if John empties an ice cube tray, we infer from the SOURCE description of an ice cube tray that he got ice from it. If John fills an ice cube tray, we infer from the CONSUMER description of an ice cube tray that he put water in it.

When objects are described as SOURCEs and CONSUMERs their descriptions are necessarily based on egocentric experience. That is, the most common experience of these objects must dominate their conceptual description. For example, most people experience a wine bottle as a SOURCE of wine. But it is conceivable that a wine bottle might be more properly conceptualized as a CONSUMER. Someone who works in a winery and fills the bottles but never drinks the stuff will conceptualize a wine bottle as more of a CONSUMER than a SOURCE. Most people share a tremendous amount of episodic knowledge. Since the inferences made in natural language processing are derived from this realm of generally shared and common experience, the knowledge representations used in natural language processing must reflect this body of common episodic knowledge.

This egocentric bias in conceptual perception has significant impact on all of our inferencing mechanisms. A teenager often views his father as a SOURCE of money. He can think of his father as someone who produces money with little consideration for the reality of earning a living and providing for a family. A person who does not smoke but has to clean up after someone who does will perceive ashtrays as a SOURCE of dirt. A person who smokes but doesn't clean up will perceive an ashtray as a CONSUMER. Environmentalists try to convince people that the earth is not an endless SOURCE of resources and that it may not be very bright to think of the ocean as a CONSUMER of all our wastes. People conceptualize the world according to their immediate experience of it and therefore operate with something less than a global view of things. A garbage can is a CONSUMER of garbage since objects which make it to the garbage can do not have to be dealt with anymore. The fact that garbage does not mystically disappear from the cosmos is an intellectual awareness which lacks the immediacy of episodic knowledge.

9.1.5 SEPARATOR and CONNECTOR

In causal chain theory, states and acts alternate; states very often enable acts and acts result in state changes [Schank 1973a]. A state is often conceptually significant because of the acts it enables or disenables. The last two Primitive Objects are designed to represent objects in terms of states which enable and disenable conceptual acts.

A SEPARATOR disconnects two regions or spatial locations with respect to a Primitive Act. SEPARATORs disenable specific acts. A CONNECTOR joins two regions or spatial locations with respect to a Primitive Act. CONNECTORs enable specific acts. Objects must assume a fixed state when being described in terms of SEPARATORs and CONNECTORs. For example, an open window is a CONNECTOR with respect to MTRANSing and PTRANSing between the inner and outer regions bounded by the window. A closed window is still a CONNECTOR with respect to visual MTRANSing but it is a SEPARATOR with respect to auditory MTRANSing and all PTRANSing. An open window and a closed window are two conceptually distinct objects. It must be understood that an open window can be transformed into a closed window, and vice-versa, but the conceptual representation of a window is ambiguous unless we know whether the window is open or closed.

A closed window and an open window are conceptually distinct objects because of the different inferences which apply to each. If a window is open we want to infer that air passes through it and that physical objects (of appropriate dimensions) can be PTRANSed through the window. But if a window is closed, none of this should be assumed. Conceptual descriptions of objects should distinguish objects in terms of valid inferences about those objects.

[Window (closed)

(a SEPARATOR with

<Disenabled = PTRANS,

$\text{MTRANS} \xleftarrow{I} \text{Speak}$>)

(a CONNECTOR with

<Enabled = $\text{MTRANS} \xleftarrow{I} \text{Eyes}$>)]

[Window (open)

(a CONNECTOR with

<Enabled = MTRANS, PTRANS>)]

```
[Road

  (a CONNECTOR with

   <Enabled = $Drive, $Bicycle, $Walk>)]


[Cut (open)

  (a CONNECTOR with                      ┌→ outside (hum0)
   <Enabled = PTRANS ←— Blood ←┤
                                         └< inside (hum0)


                                         ┌→ inside (hum0)
                PTRANS ←— Germs ←┤
                                         └< outside (hum0)>)]


[Bandaid (on)

  (a SEPARATOR with                        ┌→ inside (hum0)
   <Disenabled = PTRANS ←— Germs ←┤
                                           └< outside (hum0)>)]


[Umbrella (open)

  (a SEPARATOR with                       ┌→ hum0
   <Disenabled = PTRANS ←— Rain ←┤
                                          └< sky >)]
```

It is important to describe SEPARATORs and CONNECTORs in terms of enabled and disenabled acts. For example, a dog leash might be thought of as a CONNECTOR which fixes some locational proximity on the dog. But this is not a very useful representation in terms of potential inferences. A dog leash (when on the dog) is actually a SEPARATOR which disenables the dog from PTRANSing itself outside a fixed radius.

9.2 Applications for Object Primitives

A representational system must be judged in terms of the processes which it facilitates. While Object Primitives were motivated by problems in question answering, there appear to be many areas where they can be utilized. Four such applications will be discussed: state descriptions, script application, finding answers for where-questions, and general inference mechanisms. A brief description of a problem and an outline of the solution will be presented in each of these four areas. But first, we will look at a computer program which uses Object Primitives in order to understand.

9.2.1 COIL

COIL is a natural language processing program which implements Object Primitive descriptions in its memory representations. When COIL hears that John went into the kitchen and got some milk, a memory representation is generated which includes John getting a milk container full of milk out of a refrigerator. Had COIL heard that John went into a restaurant and got some milk, the memory representation would have involved someone serving John a glass filled with milk. COIL implements the locational specification heuristic described in 9.2.4 in order to answer certain where-questions. Inference mechanisms like the demon found in 9.2.5.1 are used by COIL during the understanding process to generate causal connections. For example, one of the inference rules used by COIL is designed to recognize that if a depleted SOURCE object is thrown away, it is thrown away because it was depleted.

Here is a sample story and Q/A session with COIL. COIL periodically flips between story understanding and question answering:

John picked up a newspaper. He went from the hall into the kitchen and got some milk.

Q: Where did John come from?

A: THE HALL

Q: Where did the milk come from?

A: THE REFRIGERATOR

But the milkcarton was empty so he threw it out.

Q: Where did the milkcarton go to?

A: THE GARBAGEBAG

Q: Why did John throw the milkcarton away?

A: BECAUSE THE MILKCARTON WAS EMPTY

He turned on the light and radio. Then he listened to music and read.

Q: Where did the music come from?

A: THE RADIO

Q: What did John read?

A: THE NEWSPAPER

Q: Why did John turn on the light?

A: SO HE COULD READ THE NEWSPAPER

Q: Why did John turn on the radio?

A: SO HE COULD HEAR THE MUSIC

HOW OBJECT PRIMITIVES WERE USED

JOHN PICKED UP A NEWSPAPER.

An expectation is aroused that John may read the newspaper. This is done by activating the Associated Scripts under the SETTING description of a newspaper.

HE WENT FROM THE HALL INTO THE KITCHEN ...

An expectation is aroused that John may prepare food. This is done by activating the Associated Scripts under the SETTING description of a kitchen.

The context of a kitchen is established from which we could derive the inference that John is in a house. This context is established by activating the Associated SETTINGs found in the SETTING description of a kitchen.

.. AND GOT SOME MILK.

There was a milk carton with milk in it. This inference is made on the basis of the OutputFrom link for milk (this link is contingent on the SETTING kitchen).

There was a refrigerator. This inference is made on the basis of the DefaultLocation link for a milk carton (this link is contingent on the SETTING kitchen).

John moved the milk carton from inside the refrigerator. This is the conceptual representation for 'getting some milk,' given the last two inferences and the RELATIONAL description of a refrigerator.

BUT THE MILKCARTON WAS EMPTY ...

There is no milk in the milk carton. The conceptual representation for the milk carton previously included a SOURCE description with Output = milk. This SOURCE description is now removed from this particular instantiation of a milk carton.

.. SO HE THREW IT OUT.

There was a garbage bag. This is derived from the GESTALT description of a kitchen.

John threw the milk carton into the garbage bag. 'Throwing something out' invokes a demon which searches for an object with an appropriate CONSUMER description. GESTALT parts of the current SETTINGs are examined. When the garbage bag is found to satisfy the requirements of the demon, it is incorporated into the conceptual representation for 'Throwing it out.'

John threw the milk carton away because it was empty. When the conceptual representation for throwing the milkcarton away is generated, a demon is triggered which tries to account for why things get thrown away. When this demon sees that the milk carton is not realizing its prototypical SOURCE description, it concludes that this is why it is being disposed of.

HE TURNED ON THE LIGHT AND RADIO.

There was a light fixture which began to emit light after John switched it on. This is represented by instantiating the SOURCE description for the light fixture.

> There was a radio which began to produce either music or some other information after John switched it on. This is represented by instantiating the SOURCE description of the radio.

THEN HE LISTENED TO MUSIC AND READ.

> The music came from the radio. This inference is immediate from the SOURCE description of the radio.
>
> John's listening to music was enabled by the radio being on. This inference is made by a demon which examines SOURCE descriptions in order to account for Enabling conditions underlying conceptual acts of ATTENDing.
>
> John's reading was enabled by the light being on. This inference is made by the same demon which connected the radio being on with listening to music.
>
> John read the newspaper. This is an instantiation of the reading script. The inference is made on the basis of the original expectation aroused at the beginning of the story when John picked up the newspaper.

9.2.2 State Descriptions

Many states are conceptually significant in terms of inferences. It is perfectly reasonable to hear:

S1: John opened the bottle and poured the wine.

But it is much more difficult to understand:

S2: John recorked the bottle and poured the wine.

S1 suggests that John poured the wine from the bottle he opened. It is impossible for John to pour wine from a bottle which he just stopped up, so S2 forces us to assume that John poured wine from some other bottle.

Being open and being closed are important states because they direct inference processes about enabled and disenabled acts. S1 sets up an enabling condition for moving liquid from the interior of the bottle, and S2 disenables movement. S1 makes sense because we can make a causal connection between opening a bottle and pouring its contents: John opened the bottle so he could pour the wine. S2 makes less sense because no such causality can be inferred: it is not clear how corking a bottle relates to pouring wine.

A conceptual representation for states should make it easy to recognize the causal relationships between states and acts. It is not enough to tag a memory token with the descriptor 'Open' or 'Closed' since these state descriptors mean different things for different objects. Open doors, open coats, open umbrellas, and open electrical switches all carry inferences which are specific to those objects. By using the Object Primitives CONNECTOR and SEPARATOR, the acts enabled and disenabled by these states are immediately apparent in the conceptual representation for these objects.

CONNECTORs and SEPARATORs can be used to describe the states 'open' and 'closed.' SOURCEs and CONSUMERs can be used to describe other states such as on, off, full, and empty. When a radio is off it is not realizing its SOURCE description. When it's on, it is. When a wine bottle is empty, it is not realizing its SOURCE description. When it's full, it is. When a garbage disposal is on, it is realizing its CONSUMER description.

Other miscellaneous states can be described in terms of Object Primitive descriptions as well. When the sun is 'up' it is realizing its SOURCE description with Output = light and heat. When John has his sunglasses 'on' they are acting as a SEPARATOR which disenables the PTRANS of light from the sun to John's eyes. When the telephone is 'broken' it is not realizing its SETTING description with the associated telephone script.

These state descriptions are useful because they specify in what way the state of a particular object relates to potential actions. Any representation of states which do not lend themselves to object-specific inferences is a weak representation. Object Primitives provide a method for state representation when the conceptual meaning of a given state varies over different objects which can assume that state.

9.2.3 Script Application

In the beginning of this chapter we alluded to a criticism of scripts which was voiced by Yorick Wilks [Wilks 1976]. The problem involved a representational weakness in scripts when a script should recognize that some objects could be appropriately substituted for other objects. For example, if John goes into a grocery store and puts items in a cardboard box or plastic bag, the script applier should be able to recognize that these are acceptable alternatives for a shopping cart or shopping basket. If John goes into the grocery store and proceeds to put items in the pockets of his coat, the script applier should realize that John is not acting in accordance with the shopping script. Other memory processes should take over at this point to deduce that John is probably up to something.

If the script applier utilized Object Primitive descriptions, there would be no difficulty in this recognition process. An object which can be used appropriately in the script is any one which has the following features in its Object Primitive description:

(a SETTING with

<Acts = HUM $\Longleftrightarrow$ ATRANS $\leftarrow$ OBJ $\xleftarrow{I}$ GRASP

HUM $\Longleftrightarrow$ ATRANS $\leftarrow$ OBJ $\xleftarrow{I}$ PROPEL[3]>)

(a RELATIONAL with

<Relationlink = Inside-of>)

(a CONNECTOR with

<Enabled = MTRANS $\xleftarrow{I}$ Eyes>)

These Object Primitive descriptions specify a container which can be carried, pushed, or pulled, and which does not obscure or hide its contents. Any appropriate substitute for a shopping basket (a pushcart, box, carrier-bag, plastic sack, etc.) will meet these specifications. Inappropriate objects like pockets, book shelves, dump trucks, plates, or hollowed-out books, will fail to meet these descriptive constraints.

OK THINGS:

Pushcart
Box (if open)
crate
carrier-bag
plastic sack
basket

NOT OK THINGS:

pocket (fails the CONNECTOR specification)
book shelf (fails the SETTING specification)
dump truck (fails the SETTING specification)
plate (fails the RELATIONAL specification)
hollowed-out book (fails the CONNECTOR specification)

If a script applier encodes descriptions of objects in terms of Object Primitives and can check Object Primitive decompositions of the objects mentioned in input sentences, there will be no recognition problem when appropriate substitutions for prototypical objects can be made.

---

[3] The Act specifiers described here have incomplete instrument fillers. The fully expanded representations would describe PTRANSes which correspond to pushing, pulling (PROPELs), or carrying (GRASP). In these conceptualizations, OBJ is a self-reference to the thing being described.

9.2.4 Locational Specification

Upon hearing the sentences:

> John was sitting in a dining car. When the train pulled out, the soup spilled.

the answers people give to:

> Q1: Where was the soup?

generally reference one of two places:

> A1: In a bowl.
> A2: On the table.

What is remarkable about this are all the answers which are not given:

> A3: On a plate.
> A4: On a floor.
> A5: In the train.

While 'In the train,' is a feasible answer to Q1, 'On a plate,' is a very strange answer. But what retrieval mechanism is responsible for recognizing that A1 and A2 are slightly better and more natural answers than A3-5? The memory representation which is generated in the course of reading the two sentences must include some relational description which places the soup inside the train. This representation should connect the soup and the train by a series of relational links specifying a path of objects which begins with the soup and ends with the train. If we generated a memory representation which used the relational links On-top-of and Inside-of, the path between soup and train would look something like:

soup (inside-of)

bowl (on-top-of)

plate (on-top-of)

tablecloth (on-top-of)

table (on-top-of)

floor (inside-of)

dining car (inside-of)

train

Given this path of objects and relational connections, how is a retrieval heuristic going to know that 'in a bowl,' and 'on the table,' are good answers but 'on a plate,' is not? One could argue that 'on a plate' is not a good answer since the soup does not rest directly on a plate. But this is not a satifactory explanation since the same reasoning could be used to argue that 'on the table,' is not a good answer (the soup does not rest directly on the table either).

If an Object Primitive decompostion is used in constructing the path from the soup to the train , the answers 'In a bowl,' and 'On a table,' can be readily preferred. A path construction utilizing Object Primitives begins with the soup, and goes to the next contingent object of contact (the bowl), but then recognizes that the bowl and plate are conceptually grouped together as parts of the GESTALT object place setting. The place setting is recognized to be part of a tablesetting, and the tablesetting would then be linked to the table by an on-top-of link. If a Part-of link were used to indicate membership in a GESTALT description, the path from soup to train would look like:

soup (inside-of)

bowl (part of)

place setting (part-of)

table setting (on-top-of)

table (part-of)

dining area (part-of)

dining car (part-of)

train

This memory representation connecting the soup to the train lends itself to a very simple heuristic for answering questions about locational specification: when a path of locational descriptions must be searched for an answer, skip over those objects which are connected by part-of links. The two objects in the above path which are not connected by part-of links are precisely the bowl and the table.

The above heuristic will not work in all cases where a choice must be made in answering a locational specification question. A complete heuristic will have to rank preferences over all of the possible relational links which may occur in a path. But the point to be made here is that a strong memory representation will result in simple retrieval heuristics. When people conceptualize the soup in the dining car they tend to clump together the bowl and the plate. Since this conceptual clumping affects answers to questions, the memory representation should reflect clumping phenomona of this sort. The notion of a GESTALT object can be used to clump objects appropriately. If further information were required (as would be the case if we asked 'What was directly under the bowl?') a memory search

---

[4] The actual construction of this path will be described in 9.2.5.2.

on the conceptual notion of a place setting could be conducted to find out that the bowl was on top of a plate. But the memory representation should not make this information predominant. We should be able to see where the plate fits in if needed, but the concept of a plate should not be as readily available as the concepts of a bowl and table.

9.2.5 Inference Mechanisms

After reading:

> John was sitting in a dining car. When the train pulled out, the soup spilled.

it is natural to answer:

> Q1: Why did the soup spill?
> A1: Because the train moved.

The causal connection between the train moving and the soup spilling must have been established at the time the two sentences were understood. If these causal connections were not continually made at the time of understanding, it would not bother us to hear:

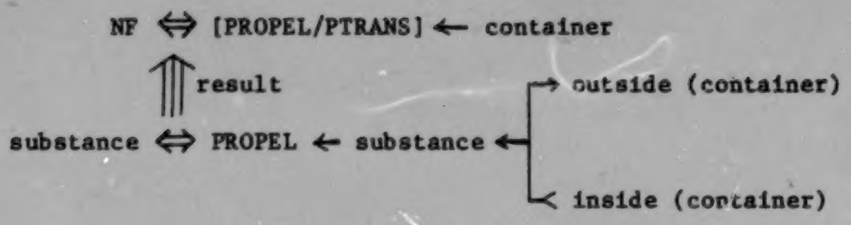
> John was sitting in a dining car. When the train pulled out, the salt shaker exploded.

This last sentence should bother us to the extent that we can't account for the salt shaker exploding. It was reasonable for the train's movement to cause the soup to spill, but it is less agreeable that the salt shaker exploded because the train moved. The causal mechanisms which try to tie conceptualizations together into causal chains cannot establish a sufficient causal link which accounts for the salt shaker exploding.

In this section we will see how Object Primitives can be used in the implementation of an inference mechanism which will recognize that the train pulling out caused the soup to spill.

9.2.5.1 A Demon

When something spills, we know that a substance of some sort has been propelled from the inside of a container to the outside. Furthermore, something must have happened to cause this occurrence since substances don't normally behave in such a manner without provocation. We don't know exactly what happened, but whatever it was, it entailed movement on the part of the container. To paraphrase the Conceptual Dependency representation of spilling, some unidentified force either PROPELled or PTRANSed a container which resulted in a substance PROPELling itself from inside of that container. In order to differentiate spilling from pouring, we will assume that the unidentified force is an unknown natural force. Since natural forces do not embody intentionality, we know that the act was not intended by anyone.


NF ⇔ [PROPEL/PTRANS] ← container
result
outside (container)
substance ⇔ PROPEL ← substance
inside (container)

In our example we want to ultimately identify the unknown force as being the train. Once the train is known to be the force at the head of the spilling conceptualization, it will be simple to answer why the soup spilled. The identification of the train relies on a capacity for recognizing the transitivity of PTRANS over containment relations.

If John puts a shirt in his suitcase and takes the suitcase to New York, we should infer that the shirt went to New York as well. But containment is not the only relation which can carry a PTRANS. On-top-of relations are also susceptible as well as various adhesive relationships. A mechanism which recognizes when PTRANS carries across objects will be essential in establishing the causal antecedent for our spilled soup. We will return to this transitivity problem shortly after a brief description of the inference mechanism.

The actual mechanism will be a demon [Selfridge 1959, Charniak 1972] which is created whenever we describe an unknown natural force PROPELling or PTRANSing an object. The purpose of the demon is to identify the unknown natural force whenever possible.

THE SPILLING DEMON:

1: is CREATED whenever we describe:

$$NF \Leftrightarrow [PROPEL/PTRANS] \leftarrow Object_1$$

2: is ACTIVATED whenever if finds*

$$X \Leftrightarrow [PROPEL/PTRANS] \leftarrow Object_2$$

*'finding' means:

a) search previous conceptualizations.

If this is not successful, go on to

b) create triggers under PROPEL and PTRANS so that a subsequently created PROPEL or PTRANS will activate the demon.)

3: once the demon is activated, it then TESTS to see if

Object$_1$ depends on Object$_2$

4: if the test succeeds then the demon ACTS by creating a result link and identifying the previously unknown force:

$$\begin{tikzcd} X \arrow[r, Leftrightarrow] & {[PROPEL/PTRANS]} & Object_2 \arrow[l] \\ X \arrow[u, Rightarrow, "result"'] \arrow[r, Leftrightarrow] & {[PROPEL/PTRANS]} & Object_1 \arrow[l] \end{tikzcd}$$

In the spilled soup problem, the conceptualization for the soup spilling creates the spilling demon (Object$_1$ = unspecified container), and the train pulling out activates it (Object$_2$ = X = train).

There are two points to be made about this mechanism. First, it does not make inferences about the PROPELling and PTRANSing of dependent objects until it has to. If no mention is made of the knife on the table, then no conceptualization involving that knife will be generated. But should an unexplained movement of the knife arise, either in subsequent text or during question answering, then this mechanism would be invoked. So this is essentially a retroactive mechanism which is summoned only when needed. It is very different from a 'forward inferencing' device which would go into action whenever any object got moved to see what other objects must have been moved as a consequence. A forward inferencing device would immediately generate conceptualizations for the movement of every object inferred to be on the train as soon as it heard that the train moved. Incidents like spilling are especially suited to retroactive processing since we don't want to generate hypotheses about everything that might possibly spill in response to any movement of any object.

The second point concerns the test part of the demon. The test does not specify a particular relationship between Object$_1$ and Object$_2$. It merely indicates some vague notion of dependence. This is intended to keep the spilling demon sufficently general so that it can account for a variety of causal chains. If we had specified an 'Inside-of' relation, then we would have had to create another demon with an 'On-top-of' test to take care of cases like:

When John bumped into the table, the vase spilled.

Since both demons would be identical except for their tests, it makes more sense to consolidate them into one mechanism and leave the testing relation a little vague. To clarify the test it may help to think of the dependence relation as being either 'On-top-of' or 'Inside-of.' In the next section we will assign a workable meaning to the general notion of dependence which is appropriate for the inference mechanism.

9.2.5.2 From Soup to Trains

In the last section we outlined roughly what sort of a causal mechanism is needed to understand that the train moving caused the soup to spill. But we have merely outlined the process. Now we will ask precisely how such a device can be implemented. In particular, we will decide what is meant by the nebulous dependency relation between objects.

In our example, we want to establish that the soup is in some sense dependent on the train. But how can this connection be made? In permanent memory there will be associative links connecting objects. For example, it is not unreasonable to have a link connecting a dining car to a train. But we can't expect to find in memory a direct associative link connecting soup and trains. To get from soup to the train a path of connecting links must be constructed. In section 9.2.4 we discussed what that path might look like, but now we must look at how that path can be constructed. The computation which creates a path must be carefully constrained by context so that the proper connection can be made without constructing every possible associative path between soup and trains. It will not help to establish a path which envisions cases of soup being shipped in a freight train.

Before we can describe the mechanism which creates a path we must know a little more about the associative links which exist between objects in permanent memory. In section 9.1.4 we saw how a wine bottle is described as a SOURCE with Output = wine. This description of a wine bottle effectively encodes an associative link from wine bottles to wine; if we examine the conceptual description of a wine bottle, we get to wine. But there should also be a link which will allow us to get from wine to wine bottles; the conceptual description of wine should lead us to a wine bottle in most contexts. In the same way, the GESTALT description of a tablesetting will take us to plates, but there should be an associative link from plates to table settings as well. And the RELATIONAL description of a bulletin board will take us to thumbtacks, but another link is needed to get from thumbtacks to a bulletin board. Before inserting reciprocal links in memory, two restrictions must be observed.

(1) Some associative links should operate in one direction only. For example, it is reasonable that a link should exist from dining cars to restaurants since a dining car can be easily recognized to be a type of restaurant. But it is not so clear that a link should exist from restaurants to dining cars. If asked to enumerate all the different kinds of restaurants there are, dining cars may not get included as an example one thinks of when considering different kinds of restaurants.

(2) Associative links may be present in some contexts but not in others. For example, in the context of eating at a table, it is reasonable that there should be a link from plates to table settings. But in the context of washing dishes, it is less likely that a plate will be perceived as part of a table setting. The organization of associative memory should be sensitive to context.

The first restriction will prevent us from postulating associative links arbitrarily whenever a need for one seems to arise. The second restriction forces us to design contingencies in conceptual memory. In the same way that the context of a story sets priorities on word senses for the parser [Riesbeck & Schank 1976], the context of a story controls descriptions of objects in conceptual memory so that associative links between objects are contingent upon context.

There are four kinds of contingent links which can exist between objects outside of their Object Primitive descriptions. The type of link which is used in any given case is determined by the object being pointed to. The four links are:

Outputfrom (points to a SOURCE)
Inputto (points to a CONSUMER)
Partof (points to a GESTALT)
Defaultlocation (points to a RELATIONAL)

The conceptual definition of a prototypical object in memory may include one or more of these associative links along with default or contingent values. A contingency in these cases is always described by either a SETTING or a script.

For example, the conceptual definition for soup describes a food which is Outputfrom a bowl during the eating script, and Outputfrom a can while in the SETTING of a kitchen or supermarket. It also has a Defaultlocation inside a pot during the prepare-food script. A bowl is described as a RELATIONAL object with Locationlink = Inside-of and as a SOURCE object during the eating script with Output = food. It is also Partof a place setting during the eating script and it has a defaultlocation inside a cabinet when in the SETTING of a kitchen. In addition to the soup, conceptual descriptions for a place setting, table setting, table, dining area, dining car, train, and passenger train are also needed to establish a path from the soup to the train. In the following descriptions, the Configurations of GESTALT objects and some contingent links have been omitted since they are not crucial to the problem at hand[5].

```
[Soup

 <OUTPUTFROM = (a Bowl during $Eating)

             = (a Can during Kitchen

                      during Supermarket)>]
```

---

[5] In these descriptions '$' is used to denote a script. Contingent specifiers which are not preceded by a '$' are SETTINGs. The contingent links which do not appear specify scripts or SETTINGs which would not be active in the context of the spilled soup problem. A few irrelevant links have been included, but not all.

```
[Bowl
 (a RELATIONAL with
  <Locationlink = Inside-of>)
 (a SOURCE during $Eating)
 <PARTOF = (a PlaceSetting during $Eating)>
 <DEFAULTLOCATION = (a Cabinet during Kitchen)>]

[PlaceSetting
 (a GESTALT with
  <Parts = Glass, Plate, Bowl, etc.>)
 <PARTOF = (a TableSetting)>]

[TableSetting
 (a GESTALT with
  <Parts = a PlaceSettings, TableCloth, etc.>)
 <DEFAULTLOCATION = (a Table)>]

[Table
 (a RELATIONAL with
  <Locationlink = On-top-of>)
 <PARTOF = (a Diningarea during $Restaurant)>
 <DEFAULTLOCATION = (a Floor)>]

[DiningArea
 (a GESTALT with
  <Parts = (Tables, Chairs)>)
 (a SETTING with
  <Scripts = $Eating>)
 <PARTOF = (a Restaurant during $Restaurant)
         = (a House during House)>]
```

```
[DiningCar
 (a SETTING with
  <Scripts = ($Eating, $Restaurant)>
  <Settings = PassengerTrain>)
 <PARTOF = (a Train during PassengerTrain)>]

[Train
 (a GESTALT with
  <Parts = (EngineCar, DiningCar, CoachCar, ...
                                during PassengerTrain)
         = (EngineCar, FreightCar, ...
                                during FreightTrain)
         = (EngineCar, CommuterCar, ...
                                during CommuterTrain)>)]

[PassengerTrain
 (a SETTING with
  <Scripts = ($Train, $Trip)>]
```

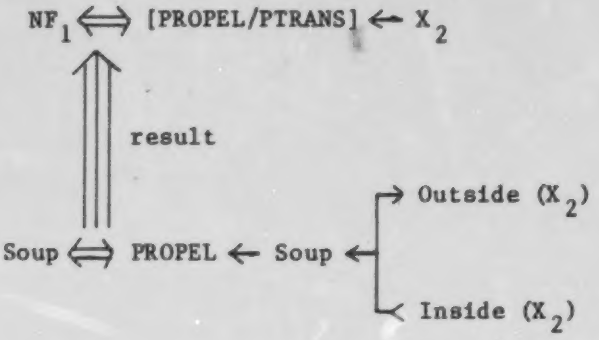
These conceptual descriptions will be utilized by our inference mechanism in order to construct a path from the soup to the train. Remember the original problem:

> John was sitting in a dining car. When the train pulled out, the soup spilled.

As soon as the first sentence is parsed, the scripts associated with a dining car are triggered. These are the eating and restaurant scripts which are found under the SETTING description of a dining car. The current context is set with the SETTING of a dining car and any other SETTINGs which are found under the SETTING description of the dining car. In this way, the current context picks up the SETTING of a passenger train in addition to the dining car SETTING. The conceptual parse of the second sentence produces two conceptualizations corresponding to (1) the train moving and (2) the soup spilling:

$$(1)\quad \text{Train} \Longleftrightarrow \text{PROPEL} \leftarrow \text{Train}$$

(2)



When the causal antecedent of the second conceptualization is generated (NF [PROPEL/PTRANS] - X) the spilling demon described in section 9.2.4.1 is created. The spilling demon then searches the previous conceptualizations to see if it can find something of the form:

$$X \Longleftrightarrow [\text{PROPEL/PTRANS}] \longleftarrow Y$$

When it finds that the conceptualization for the train pulling itself satisfies this description, the spilling demon is triggered. It must then test to see if the soup is 'dependent' on the train. Now we can define what we mean by 'dependent.' For one object to be dependent on another in the sense that this demon requires, it is necessary to be able to construct a path of associative links between the two objects.

To begin the path, we start with the soup. Looking at the conceptual representation for soup, we see that there are two Outputfrom links from soup. One is contingent on the eating script and the other is contingent upon the kitchen and supermarket SETTINGs. In the processing of our two sentences, a dining car was mentioned which added the SETTING of a dining car to the immediate context. This SETTING in turn triggered the eating and restaurant scripts. So the only associative link from soup which exists in the context of our story is the link pointing to a bowl. We therefore infer that the soup comes from a bowl.

Soup (Outputfrom during $restaurant) Bowl

Now we examine the conceptual representation for a bowl and find that it has an associative link which is contingent on the eating script. This link compels us to infer that the bowl is part of a place setting.

Bowl (Partof during $eating) PlaceSetting

The conceptual definition of a place setting gives us an associative link to a table setting. This is a default link which operates in any context.

PlaceSetting (Partof) TableSetting

The conceptual representation for a table setting specifies a default location on top of a table.

TableSetting (Defaultlocation) Table

Now when we look at the conceptual definition for a table we see that there are two paths which could be followed. A table is part of a dining area during the restaurant script, but it also has a default location on a floor.

Table (Partof during $Restaurant) DiningArea
Table (Defaultlocation) Floor

If we follow the path out from a dining area, we will get to a restaurant by a contingent Partof link. If we follow the path out from a floor, we will get to a general room by a Partof link. Neither of these branches will take us to a train. In fact, both of these branches will terminate at the concepts restaurant and room. In the current context no associative links for either restaurants or rooms exist. But this does not mean that a connecting path does not exist. We have constructed a path starting from soup and followed it out as far as it will go. Now we must start from the conceptual representation for a train and see if a path starting from a train will cross any of the concepts in the path generated from the soup.

When we went from the soup, we followed associative links between objects: Partof, Defaultlocation, Outputfrom, Inputto (the Inputto link didn't arise in this example). Starting from the other direction we will follow only those links which are internal to an Object Primitive description.

When we examine the conceptual representation for a train, we see that it is a GESTALT object whose parts are contingent on the particular type of train. The type of train we are dealing with is encoded under a SETTING constraint. In our current context, we are working within the SETTING of a passenger train. This SETTING tells us that the parts of the train are an engine car, a dining car, a coach car, perhaps a club car, and whatever other cars belong in a passenger train. So each of these parts begin a path leading from the concept of a train.

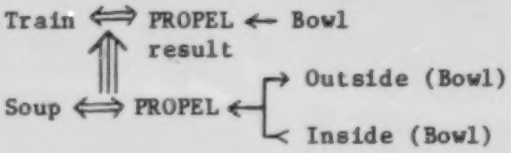
Train (with Parts during PassengerTrain)

Engine, DiningCar, CoachCar, Clubcar, etc.

Now we must examine the conceptual representations for the engine car, the dining car, the coach car, and so forth. Each of these will be GESTALT objects with various parts. So if we assume that there are about 5 parts to a passenger train and each of these parts has 10 parts of its own, then there would be roughly 50 distinct paths starting with the train. The search from this end is growing exponentially, but it stops at the second node expansion. One of the parts of the dining car will be a dining area, and this node intersects with the path which was generated starting from the soup.

DiningCar (with a Part) DiningArea

The path is therefore completed, the spilling demon decides that the soup is 'dependent' on the train, and the demon acts by placing the train as the actor in the causal antecedent for the spilled soup. The path just constructed also tells us that the container from which the soup spilled was a bowl.



The path constructing mechanism used by this inference mechanism is reminiscent of the intersection searches which Quillian proposed in his semantic networks [Quillian 1968]. But the use of a conceptual representation for objects and the notion of context-sensitive associations reduce the number of false leads dramatically.

In the path propagation originating from the soup, there are no multiple paths until the fifth generation of links. At that point the search lead to a dining area and to a floor. The search in this direction would then have two paths to follow in the sixth generation, both of which would terminate at the sixth generation. Coming from the other direction, the paths propagate exponentially, but it only takes two generations to make the intersection. In this heuristic, path propagation will always be worse coming down than going up. The path generated from the bottom follows associative links which are tightly constrained by context. The path from the top is also constrained by context, but it will fan out whenever the parts of a GESTALT have to be traced as was the case here.

One of the difficulties in implementing intersection searches is how to know when to quit and conclude there is no path. Quillian's formulation of an intersection search never specified how the search would know to give up. In our Object Primitive-based heuristic, it is feasible that paths generated from the bottom will terminate by themselves (as was the case here), and that path generation from the top could be effectively stopped after a fixed number of nodes had been touched or a fixed number of generations had been propagated.

By using associative networks which vary according to contextual constraints, the search space in an associative memory is greatly diminished. A search through associative links which is sensitive to context can be tightly directed and therefore avoid examining concepts which are completely irrelevant to the situation at hand. When someone is in a dining car and soup spills, there is no need for memory to be concerned with crates of soup in a freight car or cans of soup in a supermarket.

9.3 Conclusions

The system of Object Primitives presented here can be exploited in a variety of ways to handle many issues in knowledge representation. We have seen a few concrete applications of Object Primitives in the tasks of text understanding and question answering. Now we will step back for a moment and look at Object Primitives from a more global viewpoint.

9.3.1 Theories of Human Memory

The literature of cognitive psychology is loaded with theories of human memory. Information is not stored in memory in isolated bits and pieces; there appears to be a cohesive structure which connects everything according to some overall organization. There are three 'types' of memory which are commonly recognized within the artificial intelligence paradigm: semantic memory, episodic memory, and associative memory. Endel Tulving [Tulving 1972] proposed a distinction between semantic memory and episodic memory. The work of Quillian [Quillian 1968] is usually described as a model of semantic memory. The knowledge structures of scripts and plans [Schank & Abelson] are models of episodic memory. John Anderson and Gordon Bower [Anderson & Bower 1973] have proposed a computational model of associative memory. As researchers in natural language processing, the key question we must ask when evaluating a theory of human memory is:

How can a proposed model of human memory be used
by processes which understand and generate language?

This question defines a process model approach to the problem of understanding human memory organization.

In the formulation of Object Primitives we have developed, we are proposing an organizational structure of associative links as well as a representational system. There are two fundamental differences between our model and the memory models of Quillian or Anderson & Bower: (1) the use of contextually-dynamic memory organization, and (2) primitive decomposition in the underlying representation. In the next two sections we will discuss the advantages of these features in terms of cognitive process models.

9.3.2 Contextually Dynamic Memory

Object Primitive descriptions can be connected by contingent links which are sensitive to context. That is, an associative link may exist in one context, but not in another. In the context of a restaurant, there is an associative link between milk and drinking glasses. In the context of a store, milk is linked to a milk container. In the context of a dairy barn, milk is linked to cows. This means that the structural organization of memory is a dynamic system which alters itself as context changes.

From a processing point of view, contextual constraints are used to control searching strategies. The path generation described in section 9.2.5.2 exemplifies contextual control. The basic method is an intersection search [Quillian 1968], but with one critical innovation. Intersection searches tend to grow exponentially and cover ground which should never be touched. In a sufficiently limited knowledge domain this may not be an issue. But for a large knowledge base (as is needed for natural language processing) an exponential search strategy is not a feasible processing technique. Contextual constraints of the sort outlined in 9.2.5.2, dictate a tightly controlled direction which effectively renders the search blind to paths which cannot possibly work in the current context (like finding soup being shipped in a freight train for the soup/train problem). Contextual control has been the key to a number of problems in parsing [Riesbeck 1975, Riesbeck & Schank 1976] and inferencing [Schank & Abelson 1977]. It is therefore not surprising that contextually sensitive memory links should also be advantageous to theories of cognitive processes.

The question 'Where was the soup?' opened up an entire strategy for examining associative links in memory. Locational Specification questions can tell us about the comparative strengths of associations in memory. Grice [Grice 1975] has hypothesized a set of conversational postulates which contained some basic rules of thumb. One of them was 'Do not state the obvious.' While this is undeniably good advice, he offered no hints as to how one should go about determining the relative obviousness of things in any given situation. Here we have a representational system which can reflect some relative strengths within associative links. A Partof association is a very strong link because it is usually independent of content and is therefore liable to be a very obvious association. A stereo set is expected to have speakers, an amplifier, and turntable or tapedeck no matter where it is found - in a home, store, or recreation center. A Defaultlocation link is more likely to be contingent on the current context and therefore less obvious in the sense of general associations. We can predict milk to be in a glass, a milk container, or a cow. But these predicitions are dependent on SETTINGs like restaurants, stores, and dairy farms. Outputfrom and Inputto links are liable to be more obvious since they are less sensitive to context. Ice cube trays are typically filled with water or ice no matter where they are.

We have already seen how the relative strengths of these links can be exploited in memory retrieval. The locational specification heuristic in section 9.2.4 is based on a preference for Outputfrom and Defaultlocation links over Partof links. If we hypothesize that Outputfrom and Defaultloction links are 'less obvious' than Partof links, then this would account for the naturalness of 'In a bowl,' and 'On a table,' as answers to 'Where was the soup?' It seems promising that many memory phenomena may be explainable in terms of a model which is very fluid in its sensitivity to context. While the specific links proposed here are totally intuitive and not backed up by any pyschological data, a model for associative memory constructed in terms of Object Primitives could readily accommodate psychological data.

9.3.3 Primitive Decomposition - Why Do It?

When initially confronted with primitive decomposition as a theory of cognitive processing, a common reaction is 'Why do all that work? - It seems like so much trouble.' This impression results from a certain near-sightedness. When one begins to hypothesize specific mechanisms of inference and recognition, the advantages of a conceptual representation become apparent. But without this wider view of the processes we would like to account for, decomposition may indeed seem to be more trouble than it's worth.

There is one inference mechanism in COIL which is responsible for recognizing the enabling causality between the light being on and John reading. This very same mechanism is responsible for establishing that the radio being on enables John's listening to music. The same mechanism could find the enablement between a flashlight being on and John seeing, a car running and John smelling gas, or any other number of enablement causalities. The generality of this inference mechanism relies on the use of primitive descriptions of objects and acts. It looks for SOURCE objects and ATTEND acts.

Consider what would be needed if we did not have a level of conceptual description using primitives. Say we tried to manipulate purely lexical entities like 'light,' 'car,' 'radio,' etc. Then there is only one alternative: a specific enablement inference mechanism is needed for each pair of words which can be related by enablement. This means that each time we add a new word to our vocabulary, we have to add a set of inference mechanisms to cover that word. The number of inference mechanisms needed will grow linearly with the size of our vocabulary. This has a disastrous implication for theories of learning: the more words you know, the harder it is to learn a new one.

The semantic system proposed by Katz and Fodor [Katz & Fodor 1964] is an example of a representational system which is essentially lexical. In their representation a dictionary definition for a word is a case frame description with semantic markers constraining the entities which can appropriately fill a case slot. Case frames specify grammatical cases like 'subject' and 'object.' Semantic markers are lexical descriptions like 'higher animal,' 'physical object,' 'large,' and 'aesthetic object.' Projection rules are then responsible for checking semantic markers and making sure that words are being combined 'legally.'

There are a number of differences between the Katz & Fodor model of semantic representation and the representational system being proposed here.

> (1) Semantic markers are oriented toward finding legal word combinations; not toward representing the meaning of a sentence. Semantic markers have no way of recognizing that 'John sold the book to Mary,' and 'Mary bought the book from John,' are almost identical in meaning.

(2) Semantic markers do not encode knowledge about concepts. There are no conceptual prototypes in the theory of semantic markers. A 'ball' will have alternative definitions like [for the purpose of social dancing], or [having globular shape]. These are standard dictionary descriptions which are necessarily circular and cannot reflect conceptual knowledge.

(3) Semantic markers do not encode general knowledge about the world. A semantic marker theory will find 'The mouse chased the cat,' just as acceptable as 'The cat chased the mouse.'

(4) No theoretical structure is offered describing what semantic markers exist or if there is a taxonomy of markers. 'Pretty' would be defined to modify things which are (Inanimate V ~(Male)). The sense of 'addled' which means 'rotten' would have to take the marker (Egg). Conceptual features (inanimate, male, etc.) are being confused with associative links between words ('addled eggs').

(5) Semantic markers were conceived for the task of determining whether or not a single isolated sentence is 'grammatically acceptable,' or potentially ambiguous. This task has little to do with the cognitive processes which understand and generate memory representations for text. For example, there are no systems of inference based on semantic markers.

The major challenge facing a theory of natural language processing is the problem of inference. When are what inferences made? Where do they come from? How are they used? We need a strong theory of human inference which operates on the level of conceptual manipulation. Any theory which is formalized in terms of lexical maniplulations is doomed to failure in a system with a large (the size of an adult human) vocabulary.

If we use primitive decomposition, we need only one inference mechanism for recognizing valid enablement causalities between SOURCEs and ATTENDs. We can add new words to the system, and the original inference mechanism will automatically extend to the concepts underlying these new words. The amount of work involved in learning a new word stays constant: all we need to do to add a new word is to add a conceptual entry to the conceptual dictionaries for the parser and the generator. We do not need to add new inference mechanisms to make more inferences.

In the soup/train problem we saw an example of an inference mechanism which operates on the level of conceptual representation: the spilling demon (see section 9.2.5.1). This one mechanism will be sufficient for understanding how soup can spill in a train, how coffee can spill in a car, how a vase can spill if someone walks into the table it's sitting on, how a bucket can spill if it's dropped, and innumerably many other spilling situations. It would be impossible to achieve this sort of generality if we couldn't recognize conceptual entities like RELATIONAL objects, PROPEL and PTRANS.

In the long run, primitive decomposition is a very effective and efficient way to encode information. If information is not manipulated on a conceptual level, the processes which must be devised will be extremely specific and non-extendible when the knowledge base is increased. Primitive decomposition provides a very powerful representational system because the processes which recognize and manipulate primitive descriptions will be general and extendible.

****

The use of Object Primitives in natural language processing is an area which deserves further attention. A number of interesting theoretical problems became apparent in the course of writing COIL. But an exploration of Object Primitives and their use in general inference mechanisms is another thesis by itself. The notion of Object Primitives was introduced here mainly to illustrate how easily one can move from problems in question answering to problems of memory representation. When question answering heuristics become terribly complicated, it may be that the memory representation is inadequate. When we tried to develop heuristics to answer 'Where was the soup?' it was apparent that this answer could not be derived from the memory representation we originally had. In this way, the question answering task provides a concrete criterion for judging the strengths and weaknesses of memory representations.

CHAPTER 10

MORE PROBLEMS

10.0 Preface

In the last few chapters we have examined problems in memory representation and retrieval heuristics which are central issues in designing a question answering model. In this chapter we will turn our attention to some issues which are somewhat more peripheral. There are a number of lesser problems which must eventually be handled by any theory of question answering which claims to be complete. A few of these are outlined here. The chapter will close with a brief digression on the methodological paradigms of artificial intelligence and experimental psychology.

10.1 Consistency Checks

There are different ways that questions can fail to make sense.

Q1: Why did John hit Mary?
A1a: Who's John?
A1b: But John did hit Mary.

****
A question fails to make sense whenever
the processing of that question breaks
down for some reason.
****

Processing failures can occur at different stages within the overall process model. If a question asks about John, and the system doesn't know who John is, the interpretive processing of the question will break down during the internalization of the initial parse: no memory token for a human named John will be found. If a question asks why John hit Mary, and the system doesn't know that John hit Mary, the processing will break down during the memory search: no answer key will be found for the question concept.

When people can't answer a question because it doesn't make sense to them, they can identify what's wrong and respond accordingly. A1a and A1b are answers which respond to processing failures. These responses are generated by Consistency Check routines.

Consistency Checks are procedures embedded throughout the process model which generate appropriate responses whenever processing breaks down. There are Consistency Checks for interpretive breakdowns and failures within within the memory search. Most Consistency Checks are passive and are triggered only when a question answering process fails. For example, a Causal Antecedent Consistency Check is passive in the sense that it triggers only if the matching search fails to find an answer key. This passive process would be responsible for objecting to 'Why did John hit Mary?' when John hadn't hit Mary. But some Consistency Checks must be actively implemented. For example, an active Consistency Check for Expectational questions must be triggered whenever the question concept for an Expectational question is found in memory. 'Why didn't John hit Mary?' makes no sense if John did hit Mary.

In the computer program implementation of QUALM for SAM and PAM, priority was given to the successful processing of questions which make sense (as opposed to those which don't make sense). Therefore no Consistency Check heuristics have been implemented in QUALM. The design of Consistency Check procedures is a problem whose solutions will naturally fall out from the processes designed to handle sensible questions. That is, if we understand how to answer all of the reasonable questions which can be asked of a story, then the additional processing needed for intelligent responses to unreasonable questions will be obvious and easy to add.

10.2 Modeling Knowledge States

In Chapter Eight we discussed retrieval heuristic-based strategies for knowledge state assessment in question answering. But there we limited our observations to what can be inferred from a single isolated question. In an actual question answering dialog, people keep track of the information communicated. No human will be content to loop indefinitely:

Q: Who hit Mary?
A: John.
Q: Who hit Mary?
A: I just told you. John.

The impatience people experience when they are asked the same question twice is a by-product of processes which build and check models of other peoples' knowledge states.

In coherent question answering dialogs, questions are not answered in isolation of each other. Answers to questions are produced in accordance with what has and hasn't been said before. In the question answering dialogs which occur in courtrooms between a lawyer and a witness, the continuity of the questions asked and answers given is so tight that it is possible to see where the examination is heading; after a certain point we can predict what will be asked next (at least in dramatized courtroom dialogs). In the context of answering questions about stories, the continuity of a question answering dialog is not so important. Questions about stories are asked only for the purpose of demonstrating comprehension. The communication is therefore somewhat artificial (since the questioner knows the answers). But even in this situation we can feel disturbed by a lack of continuity:

John took a bus to New York. Then he took the subway to Leone's. He had lasagna. He took the bus back to New Haven.

Q1: How did John go to New Haven?
A1: By bus.

Q2: What did he eat?
A2: John ate lasagna.

Q3: Did John eat?
A3: Yes.

Q4: How did John go to New York?
A4: John went to New York by bus.

Q5: Did John go back to New Haven?
A5: Yes.

Q6: Where did John go?
A6: New York.

This dialog seems to be lacking direction. The questioner appears to be totally ignorant of the answers given. People would not normally ask a question like 'Did John eat?' immediately after hearing 'John ate lasagna.' The same questions could have been asked in a much more reasonably organized dialog:

Q6: Where did John go?
A6: New York.

Q4: How did John go to New York?
A4: John went to New York by bus.

Q3: Did John eat?
A3: Yes.

Q2: What did he eat?
A2: John ate lasagna.

Q5: Did John go back to New Haven?
A5: Yes.

Q1: How did John go to New Haven?
A1: By bus.

The lack of continuity is the first dialog and the apparent continuity of the second dialog are due to a process of knowledge state assessment. Whenever people answer questions, they try to keep track of what the questioner knows. That is, the answerer maintains a model of the questioner's knowledge state. Whenever a question is answered, the information communicated in that answer is incorporated in this model. If the questioner asks a question which violates our assessment of his knowledge state, we get impatient and wonder what's wrong.

Knowledge state assessment makes the difference between a system which mindlessly answers any question it hears and one which has some sense of when a question is or isn't reasonable. Underlying all Q/A dialogs is an implicit principle:

****

Questions are asked to draw attention to

gaps in the questioner's knowledge state;

questions are answered to fill those gaps.

****

This principle applies to all Q/A dialogs including the more artificial ones where the person answering must answer questions as though he were really supplying the questioner with with information the questioner didn't have.

Some task domains which involve question answering capabilities rely on a very sophisticated capacity for knowledge state assessment. For example, a system which answers (or asks) questions in a teaching situation must have a very accurate sense of what the student does and doesn't know in order to be effective [Collins 1976]. A system which answers questions in the context of technical information retrieval will be more effective if it knows how sophisticated the questioner's knowledge state is. When a question is answered on a level which is either too far above or below the questioner's level of competence, the answer will fail to communicate information.

In many cases inferences about someone's knowledge state are made on the basis of what that person has said. If you ask me whether or not all Lebesgue-integrable functions are Riemann-integrable, I will assume from your question that you know something about integration theory. More inferences are made on the basis of what I have said to you. For example, if I tell you that not all Lebesgue-integrable functions are Riemann-integrable, I might reasonably expect you to remember that and know it from then on. So alterations of knowledge state models can be made anytime anybody says anything in a Q/A dialog.

In QUALM we proposed that a Last MLOC Update (LMU) be maintained as a minimal processing technique for knowledge state assessment (see section 3.1.3.2). The LMU is a simple device for maintaining conversational continuity, and it is an intuitively valid idea since people must surely remember the conceptual content of their last statement. It is invalid insofar as it stops short of seriously modeling a memory represention for conversations. People must have memory structures of conversation which are maintained for at least some period of time. But precisely what this memory representation looks like and how it is related to the problem of knowledge state assessment is very difficult.

Knowledge state assessment is closely related to theories of conversation, overall memory organization, and issues of short term vs. long term memory. Even very simple minded notions of knowledge state models run into severe difficulties in terms of representation and overwriting. For example, suppose we wanted to create a short term knowledge state model for a given conversation. We won't even worry about integrating the information in this short term memory structure into a more permanent memory structure. The simplest structure we can propose is a list of concepts. Suppose that every time a concept is communicated to Mary by John, John adds that concept to his knowledge state list-model for Mary. Even in this short term memory model, we run into problems of updating. Consider the following dialog:

> Mary: Does Susan live in New York?
> John: No.
> Mary: Where does she live?
> John: She lives in Washington.

The following entries must be made in John's MLOC model for Mary:

1) Susan does not live in New York.
2) Susan lives in Washington.

We have a duplication of information here. If Mary knows that Susan lives in Washington, it is useless to keep a conceptualization around which encodes the fact that Mary knows Susan does not live in New York. But this reduction is only possible if a memory process of some sort determines that living in New York and living in Washington are mutually exclusive possibilities.

Suppose we could update the list-model to eliminate redundancies. Then we would have a knowledge state model which encodes the knowledge Mary acquired in the course of the conversation, but no trace is present in this model of Mary's previous knowledge states. After this dialog, John should be able to conclude that Mary did not know where Susan lived before he told her (unless he has some reason to believe she has been devious with him). Her first question indicated that Mary had some reason to believe that Susan might live in New York. If this is important to John or if it is surprising to him for some reason, he is liable to remember that Mary thought Susan lived in New York. A knowledge state model which only records those concepts communicated by the person maintaining the model, and which eliminates redundant communications, will not be adequate for representing previous knowledge states which can be inferred from a question. This raises another issue.

Should a complete running history of Mary's knowledge states be maintained? If not, how much information about previous knowledge states should be kept? Which information? What is really going on here is much more complicated than keeping lists. John is liable to remember that Mary thought Susan lived in New York only if this fact is significant or unexpected for some reason. To determine whether or not a given piece of information is significant or unexpected, a lot of memory interaction must be taking place with John's long term memory in order to pick out inconsistencies and surprises. Much more has to be known about overall memory structures and integrative processes before problems of this magnitude can be tackled in a serious manner.

10.3 Conversation Theory

Question answering dialogs are a form of conversation. There are rules of conversational structure which are used to arrive at correct interpretations and appropriate responses. The conversational continuity rules described in Chapter Three deal with very simple phenomena in conversation. To get some idea of how much more complicated conversational structure can be, let's look at an example of nested question answering dialog.

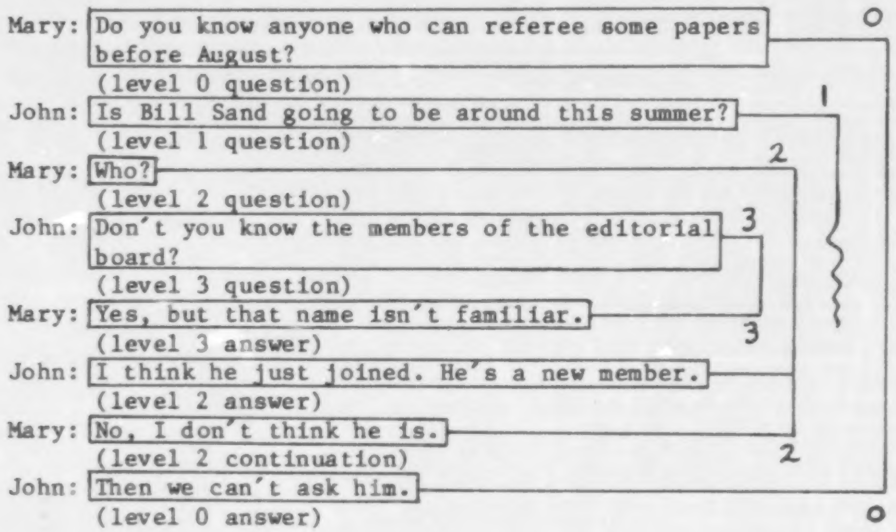
Mary: Do you know anyone who can referee some papers before August?
John: Is Bill Sand going to be around this summer?
Mary: Who?
John: Don't you know the members of the editorial board?
Mary: Yes, but that name isn't familiar.
John: I think he just joined. He's a new member.
Mary: No, I don't think he is.
John: Then we can't ask him.

In this conversation Mary disagrees with John about somebody named Bill Sand being a member of the editorial board. John seems to give in to her opinion about the matter, and says something which indicates that they shouldn't ask anyone outside of the editorial board to referee papers. The critical juncture in this conversation is when John says 'I think he just joined. He's a new member,' at which point Mary says 'No, I don't think he is.' Conversational continuity forces us to interpret Mary's statement as a reply to John. Mary seems to be saying that she doesn't think there is a new member of the editorial board named Bill Sand. John's final reply 'Then we can't ask him,' is also interpreted in terms of the reply preceding it. John seems to be saying that if the person he is thinking of is not a member of the editorial board, then they can't ask him to referee papers. The conversation makes sense as it stands and there does not seem to be any confusion about what is being said. But now look what happens when one small alteration is made:

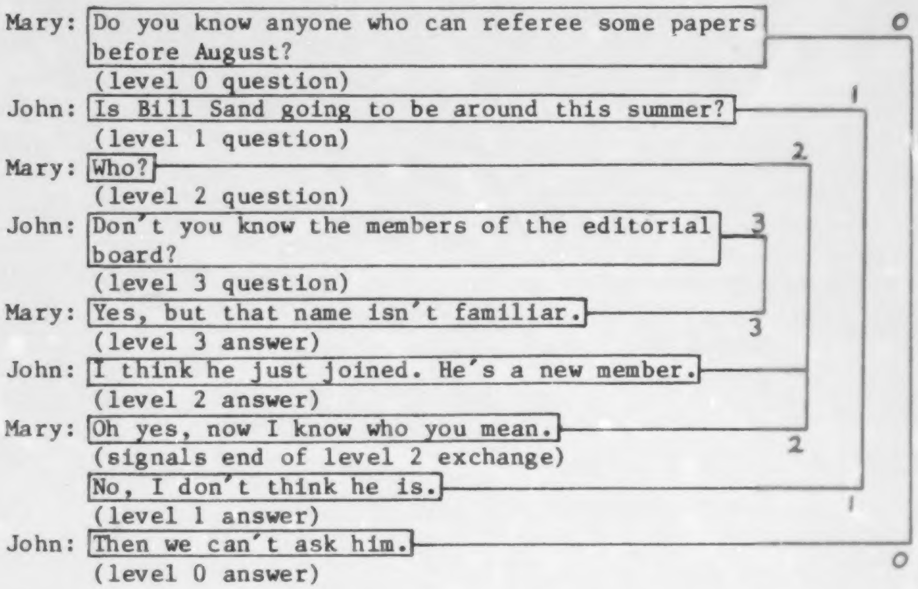
Mary: Do you know anyone who can referee some papers before August?
John: Is Bill Sand going to be around this summer?
Mary: Who?
John: Don't you know the members of the editorial board?
Mary: Yes, but that name isn't familiar.
John: I think he just joined. He's a new member.
Mary: Oh yes, now I know who you mean. No, I don't think he is.
John: Then we can't ask him.

The only change is an addition to Mary's last statement, 'Oh yes, now I know who you mean.' This declaration acts as a signal to tell John that she is resuming a suspended exchange. Without this signal her next statement 'No, I don't think he is,' is interpreted as a response to John's 'I think he just joined. He's a new member.' But with the added signal, 'Oh yes, now I know who you mean,' 'No, I don't think he is,' is interpreted as a reply to the earlier suspended question 'Is Bill Sand going to be around this summer?' The signal is critical here because the nesting of processes is three deep. People can recover from one level without signals to establish a return to the next level, but when more than one level has to be recovered, clues must be supplied to help the transitions.

Dialog Without Level Recovery Signal


Mary: Do you know anyone who can referee some papers before August?
(level 0 question)
John: Is Bill Sand going to be around this summer?
(level 1 question)
Mary: Who?
(level 2 question)
John: Don't you know the members of the editorial board?
(level 3 question)
Mary: Yes, but that name isn't familiar.
(level 3 answer)
John: I think he just joined. He's a new member.
(level 2 answer)
Mary: No, I don't think he is.
(level 2 continuation)
John: Then we can't ask him.
(level 0 answer)

Dialog With Level Recovery Signal



Mary: Do you know anyone who can referee some papers before August?
(level 0 question)
John: Is Bill Sand going to be around this summer?
(level 1 question)
Mary: Who?
(level 2 question)
John: Don't you know the members of the editorial board?
(level 3 question)
Mary: Yes, but that name isn't familiar.
(level 3 answer)
John: I think he just joined. He's a new member.
(level 2 answer)
Mary: Oh yes, now I know who you mean.
(signals end of level 2 exchange)
No, I don't think he is.
(level 1 answer)
John: Then we can't ask him.
(level 0 answer)

Without a theory of conversational structures, it is not possible to recognize when a question answering exchange is being dropped, suspended, or resumed. Questions and answers are merely units of communication within larger conversational exchanges. These exchanges are themselves subject to structural laws of combination which include nesting phenomena. Complete conceptual processing for question answering dialogs must be sensitive to these larger units of conversational structure.

10.4 What Every Lawyer Should Know

In a sophisticated model of memory representation, a story representation will evolve over time and be subject to various interference and blocking factors. Bartlett's famous experiments with 'The War of the Ghosts' [Bartlett 1932] investigated some of the ways in which memory representations deteriorate and change over time. Factors other than time can affect memory as well.

In a study by Elizabeth Loftus [Loftus 1975] subjects were shown a movie of a sports car driving down a country road. After the movie the subjects were asked a series of questions. One group was asked, 'How fast was the car going down the country road?' The other group of subjects was asked, 'How fast was the car going when it passed the barn driving down the country road?' (No barn appeared in the film.) One week later, subjects were again asked questions about the movie but without a second viewing. One of the questions at this subsequent session was 'Did you see a barn?' Only 2.7% of the first group answered Yes, but 17.3% of the second group answered Yes.

This suggests that memory representations or retrieval mechanisms can be altered by the questions which access them. In areas where a memory representation is weak or uncertain, concepts presupposed in a question can be incorporated in the memory representation. This incorporation could occur either by revising the story representation or by setting up detours for retrieval mechanisms. Presuppositions must be checked against the story representation and in the event that no contradiction is found, new information could be added to the story representation or short circuits could be set up to a more recent and slightly altered story representation generated at the time of a question answering dialog.

10.5 General Q/A

QUALM has been implemented in the task domain of answering questions about stories. But the strategies of conceptual categorization, inferential analysis, and elaboration options are all needed for a theory of general Q/A as well. All of the specific interpretive rules and content specification options outlined in this thesis are applicable to general Q/A. For example:

> Q1: Do you want a drink?
> A1: No thanks, I have to drive home.

A1 results from an application of the Correction/Explantion Option. The elaboration in A1 answers the Expectional question, 'Why don't you want a drink?'

Conversational scripts are knowledge structures for specific conversational contexts. If a Q/A dialog is embedded in a sufficiently stereotypic conversational context, these knowledge structures will guide the interpretive processing of a question. For example, a stockbroker talking to a client will understand:

> Q2: When did you sell?

to be a question about the selling price; not the time of the transaction. Stereotypic situations and relationships often provide specific predictions about what a question really means.

Theories of inference and knowledge state assessment will enable inferential analysis to understand what a questioner is really asking when a Q/A dialog occurs in the context of goal-oriented behavior which is not stereotypic. For example, if John and Mary are camping out and they are about to cook dinner, the following exchange could occur:

> John: Will you go down to the river?
> Mary: I think there's enough here.

Knowledge of general goals and plans are needed to understand that Q3 is asking, 'Will you go down to the river (to get some water (for dinner))?' Then A3 can be understood: 'I think there's enough (water) here (for dinner)).' If Mary didn't know that John wants water, that he wants it for dinner, and that the river is where you go to get water, then she couldn't have responded as she did.

The inferences needed to understand questions in general goal-oriented situations are identical to the inferences needed to comprehend text when a story describes goal-oriented behavior. If we heard:

> John needed to go down to the river before he could start dinner.

We would have to use the knowledge of scripts and plans to understand how going to the river can be causally related to preparing dinner: in the context of camping out, water exists in rivers, you frequently need water to prepare food, if you don't have something you need you must get it, and if something you need exists elsewhere you can go there to get it. This knowledge needed to understand causal connections in general text is identical to the knowledge needed to carry on general Q/A dialogs.

10.6 Psychology and Artificial Intelligence

In section 10.4 we cited evidence [Loftus 1975] that the very process of answering a question can alter a memory representation. Suppose a psychologist, confronted with the results of the Loftus experiment, wants to account for the mechanisms of human memory responsible for those results. If he works within the experimental paradigm of psychology, he will:

(1) Propose a theory describing what he thinks is going on,

(2) Design an experiment to test his theory,

(3) Run the experiment,

(4) Analyze the results to see in what ways his theory is substantiated or contradicted.

Now suppose a researcher in artificial intelligence confronts the same experimental results and would also like to account for the memory mechanisms operating. Working from within an artificial intelligence paradigm, he will:

(1) Propose a theory to describe what he thinks is going on,

(2) Write a computer program which implements this theory,

(3) Run the computer program,

(4) See if the program does what it was intended to do, and analyze the ways it fails.

In either paradigm, we are concerned with a theory of memory processes. A psychologist uses experiments to develop his theories while an AI researcher uses computer programs. In either paradigm, theories undergo continual revision and expansion. One experiment usually leads to another. And the point of writing a computer program in AI is very often just to find out how to write it better the next time. The four steps of both research paradigms describe a cyclic process: after the fourth step we go back to step one to incorporate what we have learned in a revised, extended, or totally new theory.

Does this mean that the only difference between the two paradigms is that one uses experiments and the other uses computers? Yes and no. It is the case that some problems lend themselves to experimental investigation and some don't. The experimental paradigm imposes a restriction on the kind of phenomena which can be profitably investigated. If experiments cannot be designed which isolate the variable factors of a proposed theory, the psychologist can go no further. Problems concerning human cognitive processes are difficult to study within the paradigm of experimental psychology for precisely this reason. An analogy has often been made to the effect that trying to design an experiment which will shed light on human memory processes is like trying to perform brain surgery with a hammer and chisel. What experiment can be designed to help us understand how people are able to answer simple questions like 'What's your name?' Natural language processing is a prime example of a cognitive process which slips through the net of empirical experimentation.

Natural language processing can be productively studied within the AI paradigm. If we construct a process model which is designed to account for a particular language task (e.g. question answering, summarization, translation, etc.) then we can write a computer program which implements that model. By running the program, we can see where the model is weak, where it breaks down, or where it appears competent. A program which doesn't work may not work because of technical programming errors. These can always be fixed. The interesting failures are those which occur because the process model underlying the program failed to recognize some critical problem or failed to handle some problem adequately.

When a program fails for theoretical reasons, we learn something we didn't know before. When an AI researcher wants to investigate cognitive processes in people, he uses the computer as an investigative tool which can help him see things which would otherwise be overlooked. Computers can help us study cognitive processes in the same way that microscopes help us study cell biology. Without a computer we can only guess at what's there; with a computer we're still guessing, but we at least know when we're wrong.

CHAPTER 11

PERSPECTIVE AND CONCLUSIONS

11.0 Preface

The theory of question answering proposed by QUALM is essentially a theory of natural language processing. This natural language perspective distinguishes QUALM from many other question answering systems which are motivated by information retrieval or automatic theorem proving. Many systems which attempt to answer questions phrased in natural language have been designed in two pieces: (1) a memory retrieval system, and (2) a natural language interface. Very often the interface problem is considered secondary to the retrieval system and the two subsystems are designed as if they were theoretically independent of each other.

The question answering system based on QUALM can be likewise decomposed into memory retrieval (QUALM per se) and natural language interface (parsing and generation). But in QUALM, this division distinguishes cognitive processes which are language dependent from those which are purely conceptual and independent of language. The theory of QUALM extends theories of cognitive processing which originated with the study of conceptual memory [Schank 1975], parsing [Riesbeck 1975], and generation [Goldman 1975]. QUALM therefore rests on a foundation of ideas in conceptual information processing. This orientation constitutes a fundamental departure from information retrieval and other viewpoints where natural language processing is treated as merely a 'front end' for question answering systems.

11.1 Other Q/A Systems

A number of question answering systems have been designed over the last twenty years. We will not attempt to review all of the research in computational question answering here. But it might be useful to compare QUALM with some of the more recent question answering systems in order to clarify comparative strengths and weaknesses in various approaches to question answering. The reader who is not familiar with the systems discussed here should refer to the references cited for a complete description of these research efforts.

11.1.1 Winograd (SHRDLU)

SHRDLU [Winograd 1972] was well publicized as the first computer program to understand English. The program's world knowledge was limited to a blocks world; SHRDLU simulated robotic manipulation of blocks on a table top. It could respond to requests (Please put the red pyramid on the blue block) and inquiries (Where is the red pyramid?).

From a theoretical viewpoint, Winograd presented SHRDLU as a program based on 'procedural representations of knowledge' [Winograd 1976]. Winograd maintains that the knowledge needed for a natural language processing system can be represented as procedures within that system. In SHRDLU, the understanding of 'Please pick up the block,' consists of executing a procedure which simulates picking up the block. This approach to knowledge representation is strongly dependent on the task domain Winograd chose for SHRDLU.

In a performative domain (like the blocks world), procedural representation may appear to be appropriate. But when language is understood in non-performative contexts, procedural representations fail to make sense. Story understanding is one context in which procedural understanding is not adequate. Suppose a story understanding system hears that John loved Mary but Mary didn't want to marry him. What procedure is the understanding system supposed to execute to indicate that it understood? The knowledge needed to manipulate blocks does not extend to domains where knowledge about human motives and intentionality is needed.

The question answering techniques used by SHRDLU were also limited by the task domain of the system. For example, when SHRDLU answers a 'why' or 'how' question, it consults a goal stack which was generated as a history of the system's manipulations. 'Why' questions are answered by finding the question concept in the goal stack and pushing the stack; 'how' questions are answered by finding the question concept and popping the stack. While this heuristic works in any domain where we are interested in purely procedural manipulations, it does not work in domains where some goals are more significant than others, and when goals are not the only causal antecedents for events.

> Why did John wash dishes at Leone's?

is a question with answers which describe causal antecedents but not goals:

> He couldn't pay the check.
> He had no money.
> He was pickpocketed.

Even when a goal-oriented answer is appropriate, some answers derived from a goal tree are poor because they do not take into account inferences that the questioner can make for himself:

> Why did John get on his horse?
> So he could ride it.

This is an exasperating answer which would result from simply pushing the goal stack. Some selection heuristic is needed to produce a better answer:

> Why did John get on his horse?
> So he could be close to Mary.

SHRDLU was a very impressive system since it was the first interactive computer program which responded to English input. But the theories of memory representation underlying SHRDLU have not been extended beyond its original task domain. The crucial role of knowledge representation in natural language processing has become widely recognized in recent years. But at the time SHRDLU was written, most researchers were still sorting out the real issues in natural language processing.

11.1.2 Woods (LSNLIS)

LSNLIS [Woods 1974] was a prototype natural language query system designed for accessing a large data base of technical information about the moon rock samples collected during the Apollo 11 Mission. The parser for LSNLIS is an augmented transition network which maps English questions into a parse tree. Semantic processing then translates the syntactic parse tree into a semantic meaning representation. The semantic representation for a question is actually a program expression which is directly executed for memory retrieval.

The memory representations and semantic processes used in LSNLIS were largely adequate for the technical data the system was designed to access. But there are weaknesses in the system's retrieval heuristics and semantic interpretation [Nash-Webber 1976]. For example, in order to answer:

> Do all breccias contain lanthanum?

LSNLIS successively enumerates the breccias and tests each one to see if it contains lanthanum. If all of them do, then the answer is yes. But if the data base were organized to include information like:

> All samples contain every element.
> Breccias are samples.
> Lanthanum is an element.

then a deduction process would be able to answer yes without testing each breccia sample. This problem of how to use a deductive process instead of a generate and test strategy is typical of question answering issues when a task domain involves technical descriptive information.

The question answering techniques developed for QUALM were not designed for technical information retrieval. It appears that technical information must be handled differently from information which is essentially conceptual. For example, in a technical domain it is appropriate to be concerned with deductive processess (where 'deduction' describes the generation of predicates which must be correct). But in non-technical domains, processes of inference (where 'inference' describes the generation of assumptions which may be wrong) are more useful than deductive processes. We have seen how answers to questions can be correct and still be terrible answers (see Chapters One, Three, Six, Nine). Strictly deductive processes which are sensitive to only the truth values of propositions cannot evaluate whether or not a correct answer is also an appropriate answer. The question answering problems which arise in a system like LSNLIS have very little in common with problems which arise in the context of story understanding. It is therefore useful to differentiate technical information processing from conceptual information processing.

It is too early at this stage to predict how technical and conceptual processing strategies relate to each other and precisely where the boundaries of their knowledge domains lie. A rough distinction may be made in terms of the following test: if the information encoded in a system is the sort of information which a single person (with a good memory) could be expected to remember, then the information is conceptual. Otherwise, technical. Using this guideline, stories from a weekly news magazine would belong in a conceptual information processing system, while the results of 13,000 chemical analyses belong in a technical information processing system.

11.1.3 Waltz (PLANES)

The PLANES system [Waltz 1977] is another query system which accesses a large data base of technical information. The data base for PLANES contains information about naval aircraft maintenance and flight data. It appears to be similar to LSNLIS except that the initial parse bypasses a syntactic parse tree representation in favor of a paraphrase expression of canonical phrases. This paraphrase is then fed back to the user for confirmation before an interpretive phase maps the paraphrase into a formal query language expression which is directly executed for memory retrieval.

Both LSNLIS and PLANES are examples of systems which are aimed at practical implementation in a relatively narrow knowledge domain. Design issues which would be critical for a more general system are not encountered when the task domain is sufficiently restricted. For example, relatively few words and virtually no sentences in the PLANES world are ambiguous in meaning. This simplifies parsing strategies and removes any need for inferential analysis or knowledge-based interpretation of the sort found in QUALM. The whole issue of context sensitive interpretation can be ignored when a system is designed to operate in one context only.

PLANES claims to be a data independent system. This means that the language front end of the system would not have to be substantially altered to accommodate an extended data base or a new record-based data base which is suitably constrained. (The primary alterations would entail the addition of new vocabulary.) On the other hand, the data accessing programs in PLANES would have to be substantially modified for a new data base with retrieval functions specific to that data base. The generality of PLANES is therefore defined by the type of data it accesses.

While it is difficult to compare strategies for conceptual information processing with strategies for technical information processing directly, there might be a comparison in terms of relative generality. If we draw a parallel between data bases and individual stories, then a story understanding system which has to alter its question answering techniques for each new story is analagous to a data base query system which has to alter its question answering techniques for each new data base.

QUALM has been implemented to function with story understanding systems which process stories according to theories of script and plan application. QUALM can answer questions about any story which has been understood in terms of scripts and plans. This means that new scripts can be added to SAM to increase its knowledge base, and QUALM will be able to answer questions about stories using these new scripts without any modifications to QUALM.

Since scripts and plans describe general knowledge structures, any process which relies on general properties of scripts and plans will not be affected by the addition of new scripts and plans. QUALM is designed for memory representations generated by script and plan-based story understanding systems. This gives QUALM a very powerful generality in the context of story understanding. Scripts and plans describe an extensive amount of knowledge used in text comprehension. Insofar as we can predict what scripts and plans look like in general, QUALM can be confidently applied to stories using new scripts and plans regardless of their specific knowledge content.

When a script-based retrieval technique (see Chapter Five) is required, a new script necessitates a modification to QUALM. But even then, this modification is general in the sense that it will accomodate any story understood by that script. While script and plan-specific modifications may be required, at no time should a modification to QUALM be made which is story-specific.

The PLANES system has been implemented with only one data base and does not try to substantiate claims about its generality. QUALM was originally implemented when SAM had three scripts in its data base. SAM is currently running with a total of twenty-four scripts, and no changes to QUALM have been necessary for stories based on these new scripts.

11.1.4 Scragg (LUIGI)

LUIGI [Scragg 1975] simulates stereotypic food preparation routines within the setting of a kitchen in order to answer questions like:

> What utensils would I need if I toasted bread?
> How do you make cookies?

When LUIGI receives a question, it accesses the appropriate routine (toasting bread or making cookies) and simulates the process specified. When the simulation has provided the appropriate information, the question is answered.

The theory of knowledge representation underlying LUIGI is a theory of what Scragg calls rote-oriented knowledge. From all his descriptions, rote-oriented knowledge appears to be identical to our formulation of scripts. The basic difference between LUIGI and QUALM is that LUIGI answers questions about rote-oriented procedures (scriptal activities) as a general information system, while QUALM answers questions about stories. That is, LUIGI will answer general questions about its knowledge domain; QUALM expects questions to be about some story which has been processed. LUIGI is not a story understanding system, but Scragg has suggested that rote-oriented knowledge could be used for inferencing during story understanding in the same way that SAM applies scripts during understanding to generate inferences [Scragg 1975].

The simulation processing which LUIGI implements corresponds to the proposed processing of Judgemental questions in QUALM. Judgmental questions require the questioner to make a projection about what is likely to happen on the basis of general world knowledge. What we would call a projected script instantiation, Scragg is calling a process simulation. While Scragg has described the details of processing Judgemental questions more completely than we have, QUALM covers more ground than LUIGI in terms of an overall theory. QUALM is a more comprehensive theory of question answering than LUIGI since LUIGI handles only two of the thirteen conceptual question categories which QUALM processes (Judgemental and Instrumental/Procedural). Because of this task limitation, Scragg has not developed any processing theory which corresponds to the conceptual categorization, inferential analysis, or content specification of QUALM.

11.1.5 Bobrow (GUS)

GUS [Bobrow et al. 1976] is an interactive dialog program designed to assume the role of a travel agent in a goal-oriented conversation with a client. GUS is composed of interactive modules: a morphololological analyzer, syntactic analyzer, the frame reasoner, and the language generator. The frame reasoner component of the system has received the focus of research efforts in GUS. The notion of a frame in GUS is consistent with Minsky's general formulation of frames [Minsky 1975] as prototypical template structures which can be instantiated to represent specific instances of events or entities.

GUS uses its system of travel-related frames to direct dialog and instantiate memory representations for what it is told. For example, a Date frame contains slots which can be filled specifying the month, day, year, and weekday. Other frames used by the system include Person, City, PlaceStay, TimeRange, Flight and TripSpecification frames. When GUS initiates a dialog with a client, the system asks questions in an attempt to instantiate all of its frames with information provided by the client or inferred by the system. One type of inference made by GUS is generated by default assignments for certain frame fillers. For example, the 'homeport' slot in the TripSpecification frame is never requested by GUS. This slot is automatically filled with a default assignment of Palo Alto (GUS is based in Palo Alto) unless the client specifies otherwise.

GUS is concerned with question answering from a slightly different perspective than QUALM. GUS is concerned with asking questions and understanding answers while QUALM is concerned with understanding questions and producing answers. In spite of this difference in perspective, the notion of a knowledge-specific frame system of the type used by GUS corresponds to conversational scripts in QUALM's inferential analysis (see Chapter Three).

In QUALM these predictive knowledge structures for specific conversational contexts were proposed as interpretive mechanisms for contextual understanding. In GUS these same knowledge structures are proposed as control structures which can drive dialog by utilizing the notion of procedural attachment [Winograd 1975, Bobrow & Winograd 1977]. GUS is concerned with the implementation of one specific conversational frame system while QUALM is interested in characterizing conversational scripts as a general knowledge structure which can organize contextual knowledge for question answering dialogs in a variety of contexts.

The design for frame driven dialogs proposed by GUS appears to be a promising start for mixed initiative conversation programs. GUS is still in its developmental stages. The strengths and weaknesses of its final implementation should be an instructive contribution toward a theory of conversational dialog.

11.2 Summary

The computational model of question answering proposed by QUALM is a theory of conceptual information processing based on models of human memory organization. It has been developed from the perspective of natural language processing in conjunction with story understanding systems. In the design of QUALM the following claims about question answering have been made:

(1) The processes which are specific to question answering are independent of language. This means that QUALM can operate with a parser or generator for any language without modifications to QUALM.

(2) Questions can be understood on many levels of conceptual interpretation.

(3) The level of detail inherent in a question can be determined at the time of memory search - at this time retrieval heuristics can answer the question on a level of detail appropriate to the question.

(4) It is more useful to describe answers in terms of their appropriateness rather than truth values alone. (An answer may be technically correct and still be a terrible answer).

(5) Retrieval heuristics and memory representations are two sides of the same coin. The question answering task provides concrete criteria for judging the strength of a memory representation.

(6) In the context of story understanding, some questions can be answered only by accessing expectations which were aroused at the time the story was initially read. These expectations can be reconstructed by integrating general world knowledge with the story representation.

(7) A strong taxonomy of inference based on a process model of memory will have to be developed before any general claims can be made about which inferences are made at the time of (story) understanding and which inferences are made at the time of question answering.

The ideas behind QUALM span a wide range of research areas within the field of natural language processing. In order to formulate a theory of question answering, we have been forced to confront problems in memory representation, parsing, conceptual inference, memory retrieval, knowledge structures in memory, knowledge state assessment, conversational rules and generation. There are few areas within natural language processing which are not directly related to question answering in some way. Q/A is also a natural task criterion for any language processing system which claims to understand text or any knowledge-based system which claims to be knowledgeable.

Q/A is therefore at the center of natural language processing both in terms of the natural language theory needed for question answering and in terms of the natural language systems which can be tested by question answering tasks. A complete theory of natural language processing must account for question answering phenomena. Conversely, a theory of question answering cannot be developed in isolation of natural language processing issues. Most of the research in computational question answering has treated question answering as an information retrieval problem which requires natural language processing only as a front end interface. QUALM is a theory of question answering which is founded on and which extends theories of natural language and conceptual information processing.

APPENDIX 1

THE PRIMITIVE ACTS OF CONCEPTUAL DEPENDENCY

Conceptual Dependency is a representational system that encodes the meaning of sentences by decomposition into a small set of primitive actions. When two sentences are identical in meaning, the Conceptual Dependency representations for those sentences are identical. For example, 'John kicked the ball,' and 'John hit the ball with his foot,' will have identical Conceptual Dependency representations.

Cognitive memory processes operate on the meaning of sentences, not on the lexical expression of that meaning. It follows that simulations of human cognition must rely on conceptual representations of information. Conceptual Dependency facilitates necessary recognition processes on this level of conceptual representation. For example, if memory contains an encoding for 'John bought a book from Mary,' then the processes which access memory should be able to answer 'Did Mary sell John a book?' on the basis of that encoding. This sort of recognition is trivial when 'John bought a book from Mary,' and 'Mary sold John a book,' have similar conceptual representations.

Conceptual Dependency theory is not dependent on the particular set of primitives chosen, or the number of primitives used (although the strength of a given representational system is lost if the set of primitives used is too large). The primitive acts described here define one set of primitives which have proven to be effective in the knowledge domain of general mundane world knowledge.

ATRANS

The transfer of possession, ownership, or control. ATRANS requires an actor, object, source, and recipient. E.g. 'John gave Mary the book,' is an ATRANS with actor = John, object = book, source = John, and recipient = Mary. 'John took the book,' is an ATRANS with actor = John, object = book, and recipient = John.

PTRANS

The transfer of physical location. PTRANS requires an actor, object, origin, and destination. E.g. 'John ran to town,' is a PTRANS with actor = John, object = John, and destination = a town.

PROPEL

The application of a physical force. If movement takes place because of a PROPEL, then a PTRANS occurs as well as a PROPEL. PROPEL requires an actor, object, origin, and direction. E.g. 'push,' 'pull,' 'throw,' and 'kick,' are all actions which involve a PROPEL.

MTRANS

The transfer of information. An MTRANS can occur between animals or between memory locations within a human. Human memory is partitioned into three mental locations: the CP (Conscious Processor) holds information which we are consciously aware of, the IM (Intermediate Memory) where information from the immediate context is held for potential access by the CP, and the LTM (Long Term Memory) where information is stored permanently. MTRANS requires an actor, object, source, and recipient. Sources and recipients are either animals or mental locations in a human. E.g. 'tell' is an MTRANS between people, 'see' is an MTRANS from eyes to the CP, 'remember' is an MTRANS from the LTM to the CP, and 'learn' is an MTRANS to the LTM.

MBUILD

The thought process which constructs new information from old. MBUILDS take place within the IM, receiving input from the CP and placing output in the CP. E.g. 'decide,' 'conclude,' 'imagine,' and 'consider,' are all instances of MBUILD.

INGEST

The internalization of an external object into the internal system of an animal. INGEST requires an actor, object, origin, and destination. E.g. 'eat,' 'drink,' 'smoke,' and 'breathe,' are common examples of INGEST.

EXPEL

The act of pushing an object out of the body. EXPEL requires an actor, object, origin and destination. Words for excretion and secretion are described by EXPEL. E.g. 'sweat,' 'spit,' and 'cry,' are EXPELs.

Many acts require an instrumental action on the part of the actor. The following primitive acts are used primarily as instrumental conceptualizations. Each of these acts requires an actor and object. MOVE requires an origin and destination as well.

MOVE

The movement of an animal involving some bodypart. MOVE is instrumental to actions like 'kick,' 'hand,' and 'throw.' It can also occur noninstrumentally as in 'kiss,' and 'scratch.'

SPEAK

Any vocal act. Humans usually perform SPEAKing actions as instruments of MTRANSing.

ATTEND

The act of focusing a sense organ toward some stimulus. ATTEND is almost always instrumental to MTRANS. E.g. 'see' is an MTRANS from the eye to the CP with instrument ATTEND eye to object.

GRASP

The act of securing contact with an object. E.g. 'grab,' 'let go,' and and 'throw,' each involve a GRASP or the termination of a GRASP.

****

CAUSAL CHAINS

Causal chain constructions connect individual conceptualizations by causal relationships. A fully expanded causal chain alternates events and states; events result in states and states enable events. In Conceptual Dependency there are five basic causal links.

RESULT (r)

An event 'results' in a state. This causal link can be used with any state other than mental states.

REASON (R)

Mental activity (MBUILD) can be the 'reason' for performing an action. This link joins mental events with non-mental actions.

INITIATE (I)

A state or event can 'initiate' a thought process (MBUILD).

ENABLE (E)

A state 'enables' an event.

LEADTO (L)

This causal link is used to connect two events in a causal chain representation which is not fully expanded. That is, the 'leadto' link is used to indicate that a causal chain expansion exists between two events which is not being explicitly spelled out.

CANCAUSE (C)

This link is a modification of the LEADTO link. The CANCAUSE indicates that an unspecified causal chain expansion has been left out of the causal chain representation. The difference between a CANCAUSE link and a LEADTO link is that the events and states joined by a CANCAUSE link are hypothetical.

ABBREVIATED LINKS

In the same way that the 'leadto' link is used to indicate a missing expansion, various causal links can be combined to indicate specific contractions. For example, an 'initiate-reason' (I/R) link is used to describe a state which lead to an action by means of a mental process. This link indicates that an MBUILD is implicit in the causal chain representation.

APPENDIX 2

SCRIPTS & PLANS

Scripts and plans are theoretical structures in human memory which have been proposed as models of human memory organization. A vast amount of mundane world knowledge appears to be encoded in people in the form of scripts and plans. These same constructs are being exploited as a means of organizing world knowledge in a computer. Scripts are memory units which contain information about situations or activities frequently encountered. Scripts describe the expectations involved in extremely mundane situations such as going to a restaurant, shopping in a grocery store, or stopping at a gas station. People acquire most scripts through experience and use them both operationally (as in actually going to a restaurant) and cognitively (as in understanding stories about restaurants). When you go to a restaurant, you have certain expectations about finding a table, ordering, being served, eating, getting a check, paying the check, etc. These are so ingrained that you probably don't have to spend much conscious processing time on them. Most likely you only think about them when they fail or deviate from your expectations. If you hear that John went to a restaurant and ordered a hamburger, you will infer that he ate a hamburger unless you hear something to the contrary. You weren't told that he ate a hamburger; you used your scriptal knowledge of restaurants to make the inference. While scriptal knowledge must vary from person to person according to variations of experience, there are quite a few standard scripts which will be held in common as a cultural norm. Most people have the same restaurant script since restaurants are highly standardized.

The scripts which are important for natural language processing are those which a large population holds in common. Whenever a script is shared by people, it can be referenced very efficiently. 'I went to a restaurant last night,' conveys the entire restaurant script to anyone who has that script.

Plans are used when scripts do not apply or fail to contain sufficient information. While scripts are tightly bound to well specified situations, the same plan can be invoked in a variety of settings. For example, suppose you are trying to find a friend's house in San Francisco and you have the address but you've never been there before. There is clearly no script for this situation (assuming it is novel) but you nevertheless know what sorts of things to do. You might invoke a plan which says to wander randomly until you hit the right street, but a better plan would entail knowledge acquisition. You need to find out where the street is. So you consult appropriate knowledge sources. If you have a map you look at it. If you don't have a map you might go about finding one, or you may opt for another knowledge source and try asking people if they can tell you. If you've asked ten people to no avail you might give up on finding it yourself and call your friend so he can tell you where he is or perhaps come and rescue you. The principles involved in this process are very general by nature. The same planning structures could be used for finding a particular office in the Pentagon, or finding a book in the library (without the possibility of being rescued by the book). Plans are extremely general procedures which are adaptable to a number of situations and are used when there is no standard routine to follow.

Plans and scripts are related in that plans may give birth to scripts. If I invoke the same plans for getting stores to cash my checks, and these plans are always successful, I will have a script after a while. Should my script fail at some time, I will have to revert back into planning mode. But as long as the Park Avenue address and the AMA membership card work, I will try them first. What originated as an inform/reason plan, evolved into a script due to repeated successes. For a more complete discussion of scripts and plans see [Schank & Abelson 1977].

TERMINOLOGY

A SCRIPT is a knowledge structure which is used as a predictive inference mechanism during understanding. Scripts contain knowledge about highly stereotypic situations. The scripts which are important for natural language processing are those which describe situations familiar to a large population.

SCRIPT APPLICATION is the process which accesses a script at the time of understanding in order to make predictions about what is liable to happen next. As subsequent text is processed, old predictions may be incorporated into the memory representation as inferences about things which must have occurred even if they were not explicitly stated.

SCRIPT INSTANTIATION refers to the generation of a memory representation for a specific event. The script applier instantiates scripts when it generates story representations.

DEFAULT ASSIGNMENTS are inferences made by the script applier about role bindings within a script which have not been explicilty described by the input.

SCRIPT EXECUTION refers to the actual performance of a script activity.

A PLAN is a knowledge structure which is used as a predictive inference mechanism at the time of understanding.

PLAN APPLICATION is the process which accesses plans at the time of understanding in order to make predictions about what is liable to happen next.

PLANNING STRUCTURES refer to those constructions in a memory representation which were generated by plans.

APPENDIX 3

STORY REPRESENTATIONS

In this overview we will describe the story representations generated by SAM and PAM. But we will not attempt to describe how these representations are created during the understanding process. For a discussion of SAM's understanding processes see [Cullingford 1977] and for a description of the PAM system see [Wilensky 1976]. In general we can say that both systems are based on predictive mechanisms which make many inferences at the time of understanding. All substantiated inferences made during understanding are incorporated in the story representation, and once the story representation has been created there is no record of which conceptualizations in the story representation were explicitly stated in the story and which were inferred. Most of the questions which can be answered about a story are answerable on the basis of the story representation alone without additional inferencing or reasoning. For a theoretical discussion of memory and inferencing in story understanding, see [Abelson & Schank 1977].

CAUSAL CHAINS

The causal chain level of representation encodes a chronology of events and states describing everything which happened in the course of the story. The individual conceptualizations in this chain are joined by causal links according to the syntax of causal chains [Schank 1973a]. Basically, a causal chain is a string of alternating states and acts, with the causal links ENABLE, RESULT, REASON, INITIATE, CANCAUSE, and LEADTO joining them. Causal chains tend to be fairly linear, but there are times when a single conceptualization will have multiple antecedents or consequents.

One way to think about a causal chain representation is to imagine a movie of the story and to record each event as it occurs. For example, in a story which describes John going to a restaurant, there are dozens of events which must involve John between the time he enters the restaurant and the time he leaves. Each one of these events goes into the causal chain. If there is more than one character in a story, there will be a point of view adopted for the causal chain representation. If John goes to a restaurant, the causal chain will be dominated by events which involve John directly. We want to keep the camera on John for the most part.

To give you some concrete sense of causal chain representations, let's look at a story which SAM has understood:

> John went to New York by bus. On the bus he talked to an old lady. When he left the bus he thanked the driver. He took the subway to Leone's. On the subway his pocket was picked. He got off the train and entered Leone's. He had some lasagna. When the check came he discovered

> he couldn't pay. The management told him he would have to wash dishes.

The following paraphrase for this story was generated by SAM by translating into English all of the conceptual acts found in the causal chain representation for this story [Schank et al. 1975]. The actual causal chain has states between consecutive acts, but these were not included in the paraphrase. Notice how single sentences from the original story are broken down into sequences of conceptual acts. Also notice how conceptualizations have been inferred from the story in order to form complete causal chains between two input conceptualizations:

> John went to a bus stop. He waited at it a few minutes. He entered a bus. The driver got the ticket from John. He went to a seat. He sat down in it. While John was on the bus an old lady and John talked. The driver took John to New York. He went to the driver. When getting off the bus John thanked the driver. He got off it. He entered a station. He put a token in the turnstile. He went to the platform. He waited at it a few minutes. He entered a subway car. A thief went to John. He picked John's pocket. He went. John went to a seat. He sat down in it. The driver took John to Leone's. He left the subway car. He left the station. He entered Leone's. He looked around inside it. He saw he could go to a table. He went to it. He sat down in the seat. He ordered some lasagna. The waiter indicated to the chef John would like him to prepare something. The chef prepared the lasagna. The waiter got it from the chef. The waiter went to the table. He served the lasagna to John. He ate it. He became full. He asked the waiter for the check. John got it from the waiter. John read the check. John discovered he was unable to pay the check. He indicated to the waiter he was unable to pay the check. The management told John he would have to wash dishes. He entered the kitchen. He washed dishes. He left Leone's.

Some of the inferences made here could be wrong. For example, there is no way of knowing whether John had his pocket picked before or after he sat down on the subway. For that matter, we don't really know that John sat down on the subway. In the causal chain representation some inferred conceptualizations are tagged according to their relative certainty [Cullingford 1977]. But the basic notion behind the causal chain level of representation is a chronology of states and acts describing everything which must have happened.

SCRIPT STRUCTURES

Script structures constitute a higher level of representation. They encode a rough bird's eye view of the story. This level of representation is most important when more than one script is referenced in the course of a story. Script structures describe how various scripts relate to each other. For example, a general trip script will contain various travel scripts (nested or sequentially ordered), as well as scriptal descriptions of the destination activities. Scripts are generally related sequentially or by nesting, and some scripts (like the trip script) predict scriptal relations within themselves. The scriptal structure for the Leone's story looks like:

$TRIP1	GOING	$BUS1 $SUBWAY1 ($PICKPOCKET1)
	DESTINATION	$RESTAURANT1
	RETURNING	NIL

Each pointer in a script structure points to a particular instantiation of a script. $BUS1 in the going part of $TRIP1 refers to that part of the story which involved a bus ride to New York. This instantiation of the bus script binds John as the main actor and includes him thanking the driver and talking to an old lady. Had the story included John going back home by bus, a second instantiation of the bus script would appear under the returning part of $TRIP1. While most of the scripts which appear in the Leone's story are sequentially related, some are nested within others. All scripts in the Leone's story are part of the trip script, and the pick-pocketing episode is nested within the subway ride.

Script stuctures allow access to script-related information which is stored independent of the causal chain representation. Each pointer in a script structure allows access to the conceptualization summarizing that script instantiation, any events which occurred during that script instantiation which were recognized as being particularly unusual or interesting, and any role bindings for that particular script instantiation. This ability to extract the most important aspects of a script instantiation is crucial in many retrieval tasks.

PLANNING STRUCTURES

In order to understand some stories knowledge is needed about human goals and strategies for achieving these goals. Suppose John saves Mary from a dragon and she marries him. This makes sense as a unified story only because we infer that the dragon intended to harm Mary, John didn't want Mary to get hurt, and Mary was grateful to John after he rescued her. Suppose instead that Mary married John after he fed her beloved pet poodle to a lion. This story is much harder to understand because it violates rules about human behavior and mental states. For Mary to marry John we expect that Mary loves or likes John. But if John destroyed a pet which she liked, we expect Mary to dislike John. If Mary marries John after his cruel behavior, we are forced to assume either that we are missing some critical piece of information (although it is hard to imagine what), or that Mary and John are into some sort of strange sadist/masochist relationship. The causality between John turning Mary's pet poodle into cat food and Mary marrying John is very difficult to account for.

In order to understand stories on this level of human motivation and behavior, plans must be invoked in the understanding process. The story representation generated must incorporate plan-related information. PAM creates story representations which include both a causal chain level of representation and a planning structure representation. To see what kinds of plan-related inferences are made at the time of understanding, consider the following story:

> John loved Mary but she didn't want to marry him. One day, a dragon stole Mary from the castle. John got on top of his horse and killed the dragon. Mary agreed to marry him. They lived happily ever after.

In the course of understanding this story, PAM makes the following inferences:

> John wanted to marry Mary
> Mary was endangered by the dragon
> John learned that the dragon had kidnapped Mary
> John wanted to save Mary from the dragon
> John rode his horse to where Mary was
> Mary became grateful to John for rescuing her
> John and Mary got married.

In addition to these inferences, there are plan-related causal links which inter-relate these inferences. For example, the story representation encodes the fact that Mary married John because she felt grateful to him (for rescuing her), John got on his horse in order to get to where Mary was, being where Mary was enabled John to rescue her, and John knowing that the dragon had kidnapped Mary initiated a thought process in John which ended in his deciding to save Mary. The causalities connecting these conceptualizations go beyond the chronological causality found in causal chains. These relational links are concerned more with why people do what they do than the physical chain of events which occur in the course of a story.

In addition to plan-related inferences, PAM generates a causal chain representation which encodes the events of the story as they occurred. To get a sense of the causal chain representation generated by PAM, let's look at a paraphrase based on the causal chain for the dragon story. This paraphrase was generated in the same way that the long paraphrase for the Leone's story was created by SAM. Each conceptual act in the causal chain has been translated to English, leaving out all of the intervening states with the exception of emotional states:

> John was in love with Mary. She did not want to marry him. A dragon took her from the castle. He learned that the dragon had taken her from the castle. He mounted a horse. It took him to her. He killed the dragon. She was indebted to him. She told him she was going to marry him. He married her. He and she were happy thereafter.

This paraphrase gives some sense of how the story is understood on the causal chain level alone. The inferences about goals and plans with the causal relationships connecting them exist outside of the causal chain representation but are connected to conceptualizations in the chain. The level of planning structures exists in story representations as a sort of overlay which lies on top of the causal chain representation. For a description of how PAM generates causal chains and script structures during understanding see [Wilensky 1976].

BIBLIOGRAPHY

Anderson, J. R. and Bower, G. H. (1973) Human Associative Memory. John Wiley and Sons, New York.

Bartlett, R. (1932). Remembering: A Study in Experimental and Social Psychology. Cambridge University Press, London.

Bobrow, D. G., Kaplan, R., Kay, M., Norman, D., Thompson, H., Winograd, T. (1976). GUS, A Frame-Driven Dialog System. Xerox Palo Alto Research Center, Palo Alto, Calif. (to appear in Artificial Intelligence).

Bobrow, D. G. and Winograd, T. (1977). An Overview of KRL, a Knowledge Representation Language. Cognitive Science vol.1, no.1.

Bower, G. H. (1976). Comprehending and Recalling Stories. Div. 3 Presidential Address, Washington: American Psychological Association, Sept. 6. 1976.

Bransford, J. D. and Franks, J. J. (1971). The Abstraction of Linguistic Ideas. Cognitive Psychology vol.2 pp. 331-350.

Carbonell, J. (1977). Ideological Belief System Simulation. Department of Computer Science, Yale University. New Haven, Ct. (submitted to the Fifth International Joint Conference on Artificial Intelligence).

Charniak, E. (1972). Towards a Model of Children's Story Comprehension. (thesis) AITR-266 M.I.T. Cambridge, Mass.

Charniak, E. (1975a) Organization and Inference in a Frame-like System of Common Knowledge. Proceedings from Theoretical Issues in Natural Language Processing. Cambridge, Mass.

Charniak, E. (1975b). A Partial Taxonomy of Knowledge about Actions. Proceedings of the Fourth International Joint Conference on Artificial Intelligence. Tbilisi, USSR.

Collins, A. (1976). Processes in Acquiring Knowledge. Schooling and the Acquistion of Knowledge. Eds: Anderson, Spiro, and Montague. Erlbaum Assoc. Hillsdale, N.J.

Cullingford, R. E. (1975). An Approach to the Representation of Mundane World Knowledge: The Generation and Management of Situational Scripts. American Journal of Computational Linguistics. Microfiche #44.

Cullingford, R. E. (1976). The Uses of World Knowledge in Text Understanding. Proceedings of the Sixth International Conference on Computational Linguistics. Ottawa, Canada.

Cullingford, R. E. (1977). *Organizing World Knowledge for Story Understanding by Computer*. (thesis) Department of Engineering and Applied Science. Yale University, New Haven, Ct.

Feigenbaum, E. A. (1963). The Simulation of Verbal Learning Behavior. *Computers and Thought*. Eds: Feigenbaum and Feldman. McGraw Hill, New York.

Goldman, N. M. (1974). Computer Generation of Natural Language from a Deep Conceptual Base. Stanford Artificial Intelligence Laboratory Memo AIM-247. Stanford University, Stanford, Calif.

Goldman, N. M. (1975). Conceptual Generation. *Conceptual Information Processing*. Ed: Schank. North-Holland, Amsterdam.

Grice, H. P. (1975). in *Explorations in Cognition*. Eds: Norman and Rumelhart. W. H. Freeman and Co. San Francisco.

Katz, J., and Fodor, J. (1964). The Structure of a Semantic Theory. *The Structure of Language*. Eds: Fodor and Katz. Prentice Hall. Englewood Cliffs, New Jersey.

Lehnert, W. (1977). Human and Computational Question Answering. *Cognitive Science* vol.1, no.1.

Loftus, E. F. (1975). Leading Questions and the Eyewitness Report. *Cognitive Psychology*. (7) pp. 560-572.

Minsky, M. (1975). A Framework for Representing Knowledge. *The Psychology of Computer Vision*. Ed: P. H. Winston. McGraw-Hill, New York.

Nash-Webber, B. (1976). Semantic Interpretation Revisited. A.I. Report No. 48., Bolt Beranek and Newman Inc. Boston, Mass.

Norman, D. (1972). Memory, Knowledge, and the Answering of Questions. Center for Human Information Processing Memo CHIP-25. University of California at San Diego.

Norman, D. A. and Rumelhart, D. E. (1975). *Explorations in Cognition*. W. H. Freeman and Co. San Francisco.

Quillian, M. R. (1968). Semantic Memory. *Semantic Information Processing*. Ed: Minsky. MIT Press, Cambridge, Mass.

Rieger, C. (1975a). Conceptual Memory. *Conceptual Information Processing*. Ed: Schank. North-Holland, Amsterdam.

Rieger, C. (1975b). The Commonsense Algorithm as a Basis for Computer Models of Human Memory, Inference, Belief and Contextual Language Comprehension. *Proceedings from Theoretical Issues in Natural Language Processing*. Cambridge, Mass.

Riesbeck, C. (1975). Conceptual Analysis. Conceptual Information Processing. Ed: Schank. North-Holland, Amsterdam.

Riesbeck, C. and Schank, R. (1976). Comprehension by Computer: Expectation-Based Analysis of Sentences in Context. Research Report #78. Department of Computer Science, Yale University, New Haven, Ct.

Schank, R. C. (1972). Conceptual Dependency: A Theory of Natural Language Understanding. Cognitive Psychology. 3(4) pp. 552-631.

Schank, R. C. (1973a). Causality and Reasoning. Technical Report #1. Istituto per gli Studi Semantici e Cognitivi, Castagnola, Switzerland.

Schank, R. C. (1973b). Identification of Conceptualizations Underlying Natural Language. Computer Models of Thought and Language. Eds: Schank and Colby. W. H. Freeman and Co. San Francisco.

Schank, R. C. (1974a). Adverbs and Belief. Lingua. 33(1). pp. 45-67.

Schank, R. C. (1974b). Understanding Paragraphs. Technical Report #6. Istituto per gli Studi Semantici e Cognitivi, Castagnola, Switzerland.

Schank, R. C. (1975a). Conceptual Information Processing. North-Holland, Amsterdam.

Schank, R. C. (1975b). The Structure of Episodes in Memory. Representation and Understanding: Studies in Cognitive Science. Eds: Bobrow and Collins. Academic Press, New York.

Schank, R. C. and Abelson, R. P. (1975). Scripts, Plans, and Knowledge. Proceedings of the Fourth International Joint Conference on Artificial Intelligence. Tbilisi, USSR.

Schank, R. C. and Abelson, R. P. (1977). Scripts, Plans, Goals and Understanding. Lawrence Erlbaum Assoc., Hillsdale, N.J.

Schank, R. C. and Colby, K. M. (1973). Computer Models of Thought and Language. W. H. Freeman and Co. San Francisco.

Schank, R. C. and Yale A.I. Project (1975). SAM -- A Story Understander. Research Report #43. Department of Computer Science, Yale University, New Haven, Ct.

Scragg, G. W. (1975). Answering Questions about Processes. Explorations in Cognition. Eds. Norman and Rumelhart. W. H. Freeman and Co. San Francisco.

Selfridge, O., (1959). Pandemonium: A Paradigm for Learning. Proceedings of the Symposium on Mechanisation of Thought Processes. Eds: Blake and Uttley. H. M. Stationary Office, London.

Shortliffe, E.H. (1974). MYCIN: A Rule-Based Computer Program for Advising Physicians Regarding Antimicrobial Therapy Selection. (thesis) Memo-AIM251. Stanford Artificial Intelligence Laboratory. Stanford University. Stanford, Calif.

Tulving, E. (1972). Episodic and Semantic Memory. Organization of Memory. Eds: Tulving and Donaldson. Academic Press, New York.

Waltz, D. (1977). An English Language Question Answering System for a Large Relational Data Base. Coordinated Science Library, University of Illinois, Urbana, Ill. (submitted to the Communications of the ACM).

Wilensky, R. (1976). Using Plans to Understand Natural Language. Proceedings of the Annual Conference of the ACM. Houston, Texas.

Wilks, Y. (1976) De Minimis, or the Archaeology of Frames. Department of Artificial Intelligence, University of Edinbourgh, Edinbourgh, Scotland.

Winograd, T. (1972). Understanding Natural Language. Academic Press, New York.

Winograd, T. (1975). Frame Representations and the Declarative-Procedural Controversy. Representation and Understanding. Eds: Bobrow and Collins. Academic Press,